Machine Learning Survey

3rd OBELICS workshop

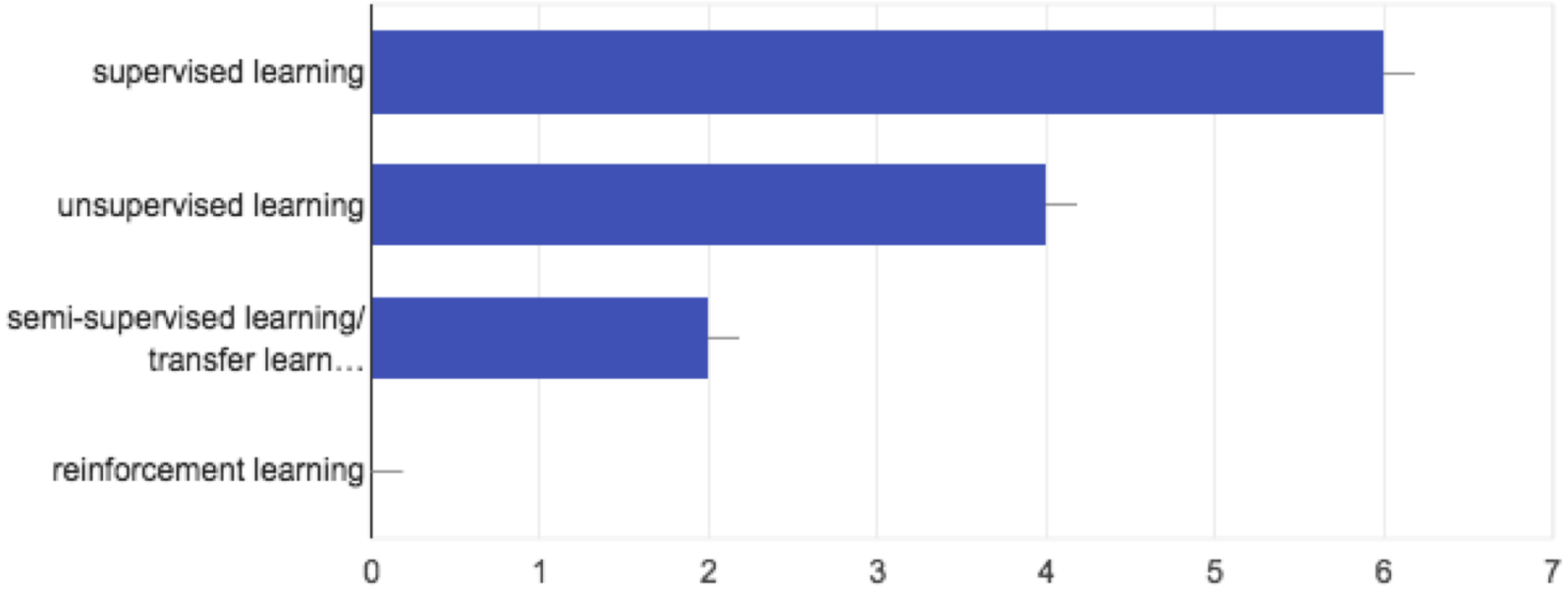
Cambridge, October 2018

What experiment do you work on?

- SKA Design and Development
- LSST
- EST
- Virgo
- KM3NeT
- ATLAS
- CTA

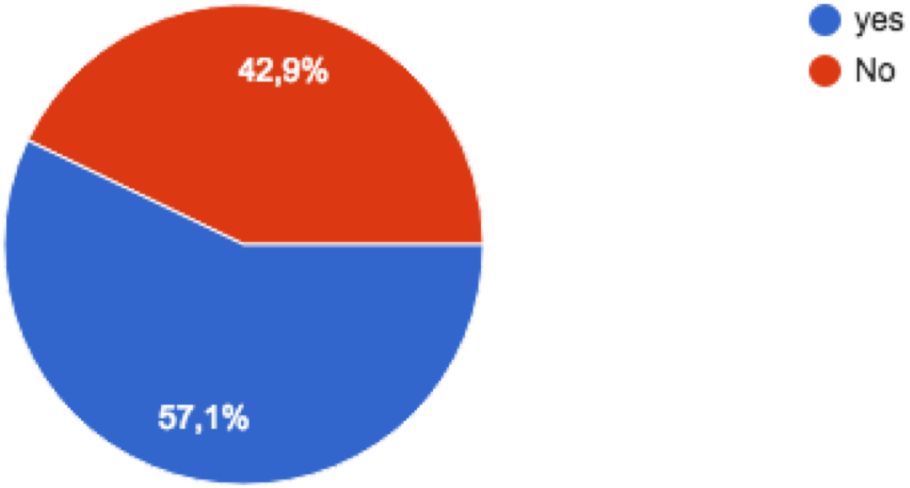
Do you use

7 réponses



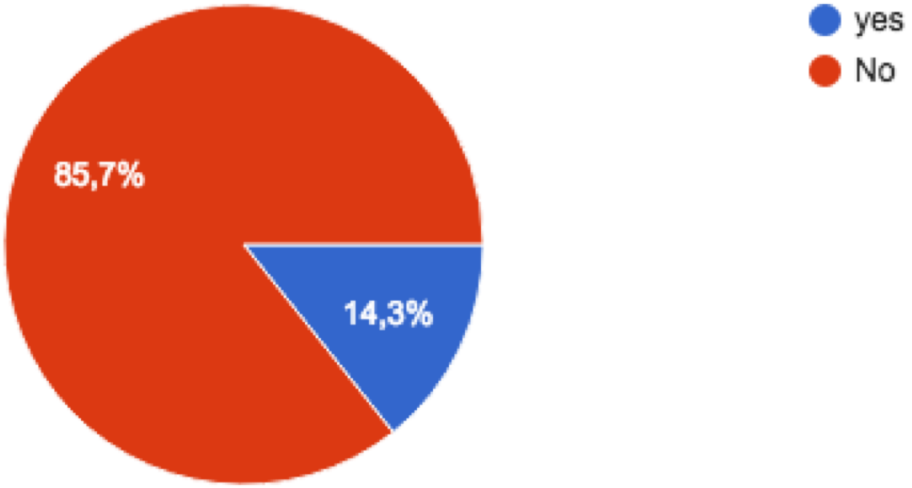
Is machine learning used in production?

7 réponses



Is deep learning used in production?

7 réponses

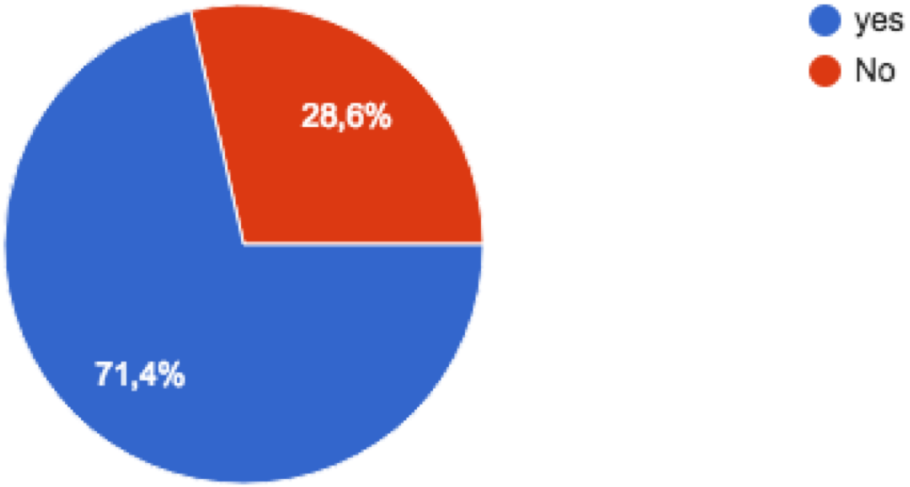


Collaborations

- Common data format ?

Did you need to adapt the data (e.g. resampling) and/or the machine learning techniques (e.g. convolution kernels) to use standard frameworks?

7 réponses



What data format do you work with?

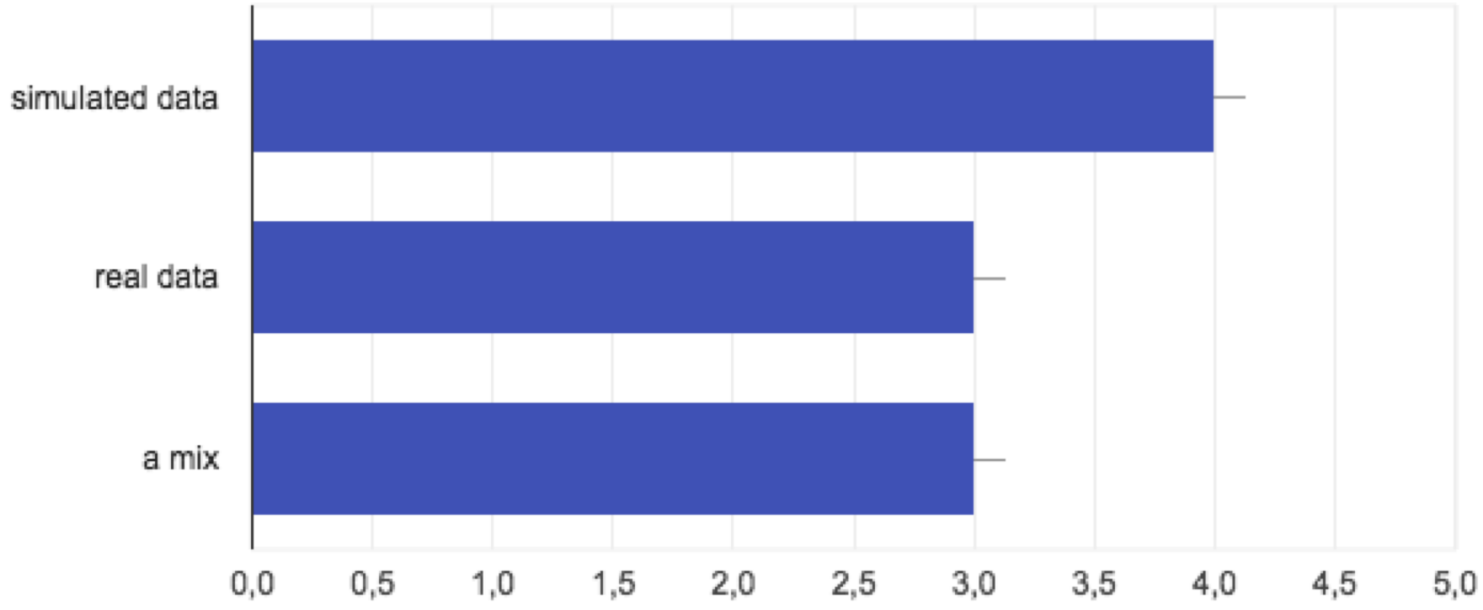
- CSV, ARFF, PSRFITS, Filterbank, custom reduced data products, e.g. feature data only
- FITS files
- Images
- frame files, hdf5
- ROOT trees, numpy array
- ATLAS specific, based on ROOT. Often translated to more ML friendly formats like numpy arrays stored in HDF5 files

Accessing the data

- Data preparation/extraction takes a large part of the processing time
- Data Format
  - Common data format
  - Significant data sharing/format issues inside collaborations already
- Eliminate dependencies from the experiment-related software.

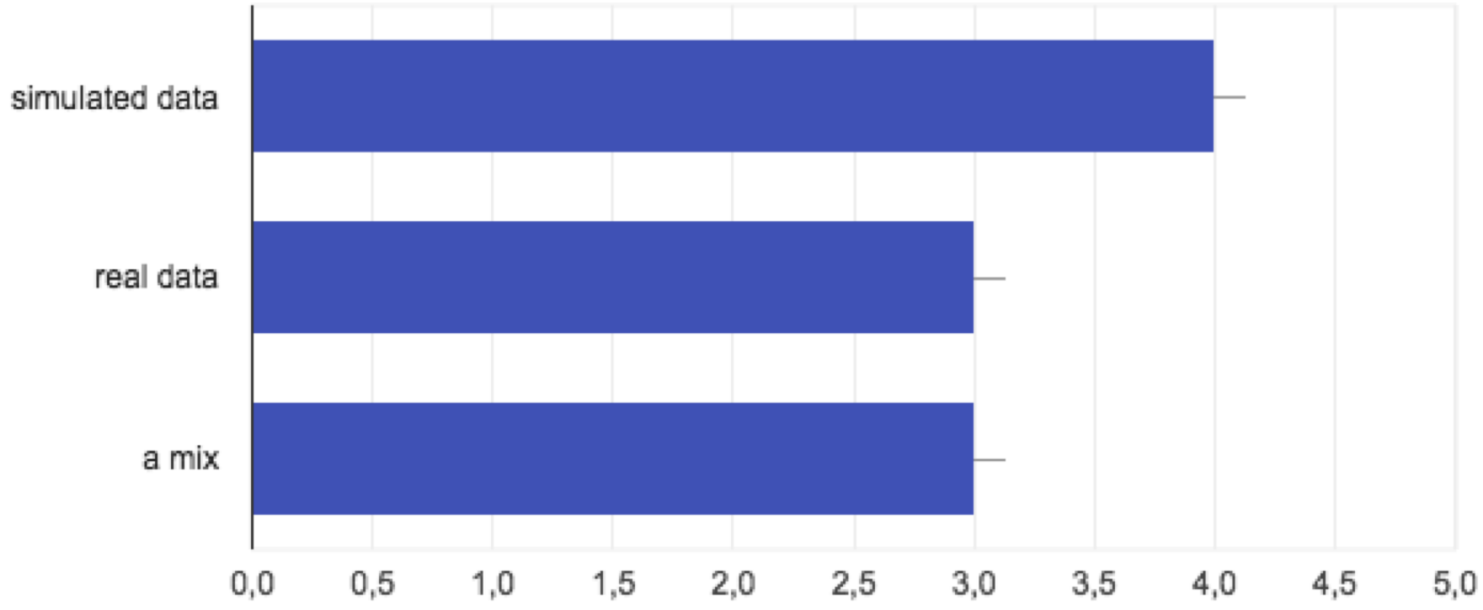
Do you train on

7 réponses



Do you infer on

7 réponses



Collaborations

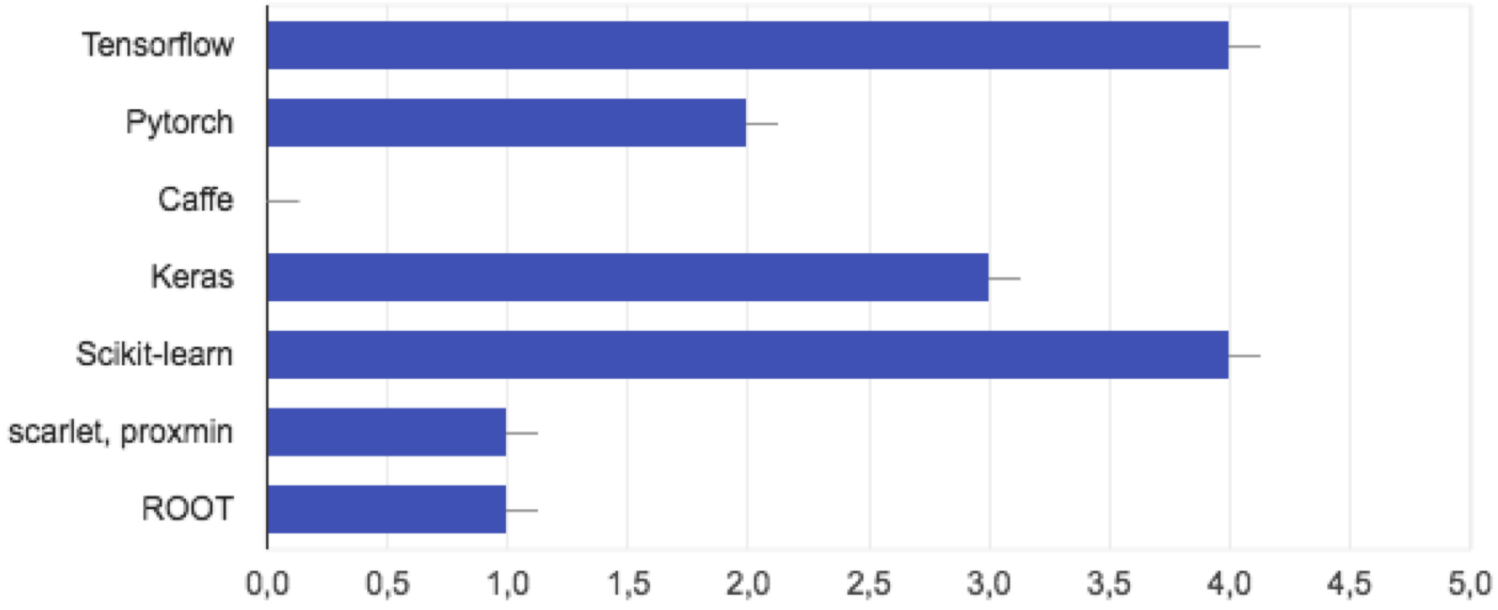
- Common data format ?
  - **Open data !**

Collaborations

- Common data format ?
  - **Open data !**
- Common tools?

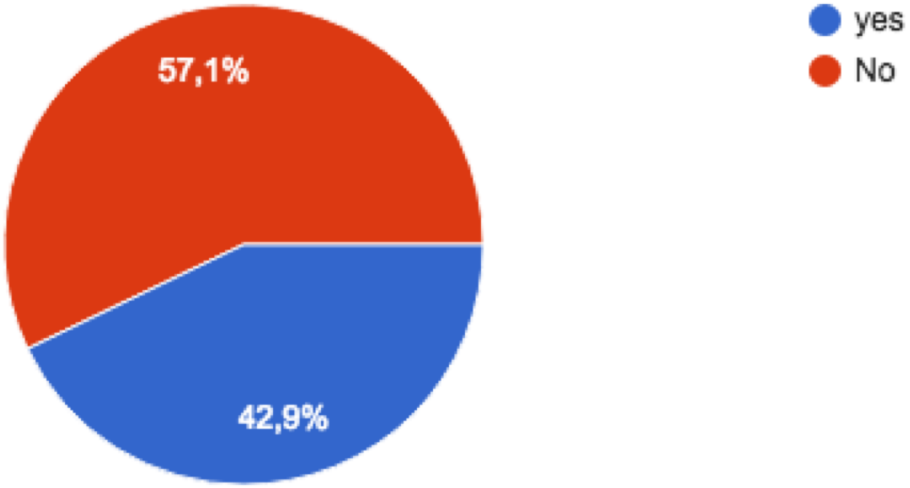
What framework/library do you use

7 réponses



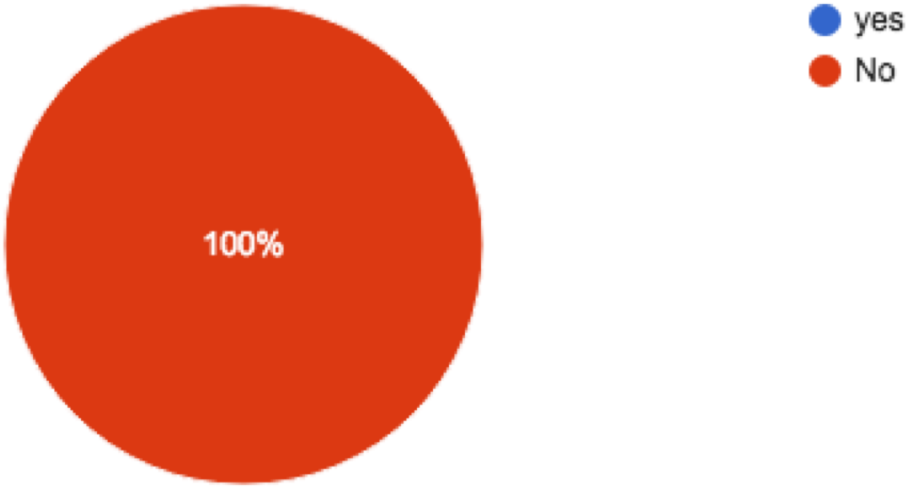
Do you develop in a larger framework/pipeline?

7 réponses



Is the framework/library imposed by your collaboration?

7 réponses



How do you interface science tools with machine learning frameworks?

- Custom / hand-made code
- extract the information needed to process data with the ML frameworks (e.g. extract data from ROOT trees using root_numpy or pyroot), create numpy arrays and adapt the data format to the framework and to the type of model decided to use
- Python interfacing
- In production we use C++, and have to take trained models , save them, and re-build them in C++ with custom tools for production time inference

Machine learning is a lot about bookkeeping. How do you deal with this? How do you compare the different tests/experiments your run?

- Jupyter notebooks.
- QA plots and statistics
- I have still not found an easy solution
- Meticulous book keeping and saving different models with metadata to store this information
- Containerization is beginning to help, since we can save containers with different models
- In production, releases are stored in databases and are updated relatively rarely, so the information for a fixed production release is heavily documented

Collaborations

- Common data format ?
  - **Open data !**
- Common tools?
  - open-source https://www.comet.ml/ ?
  - Visualisation ?
  - Data exploration ?
- Sharing knowledge
  - What didn't work
  - What to expect from this or this model/algorithm
  - Exchange platform ? (github?)
  - Astro machine learning conference ?

Collaborations

- Common data format ?
  - **Open data !**
- Common tools?
  - open-source https://www.comet.ml/ ?
  - Visualisation ?
  - Data exploration ?
- Sharing knowledge
  - What didn't work
  - What to expect from this or this model/algorithm
  - Exchange platform ? (github?)
  - Astro machine learning conference ?
- Pilot project ?
- Inputs for Rob Lyon:
  - Imbalanced classifiers
  - Develop best practices together (DevOps for ML?)