

Open Source for (Data) Science

Andreas Müller

Associate Research Scientist
Columbia University
Scikit-learn Technical Committee

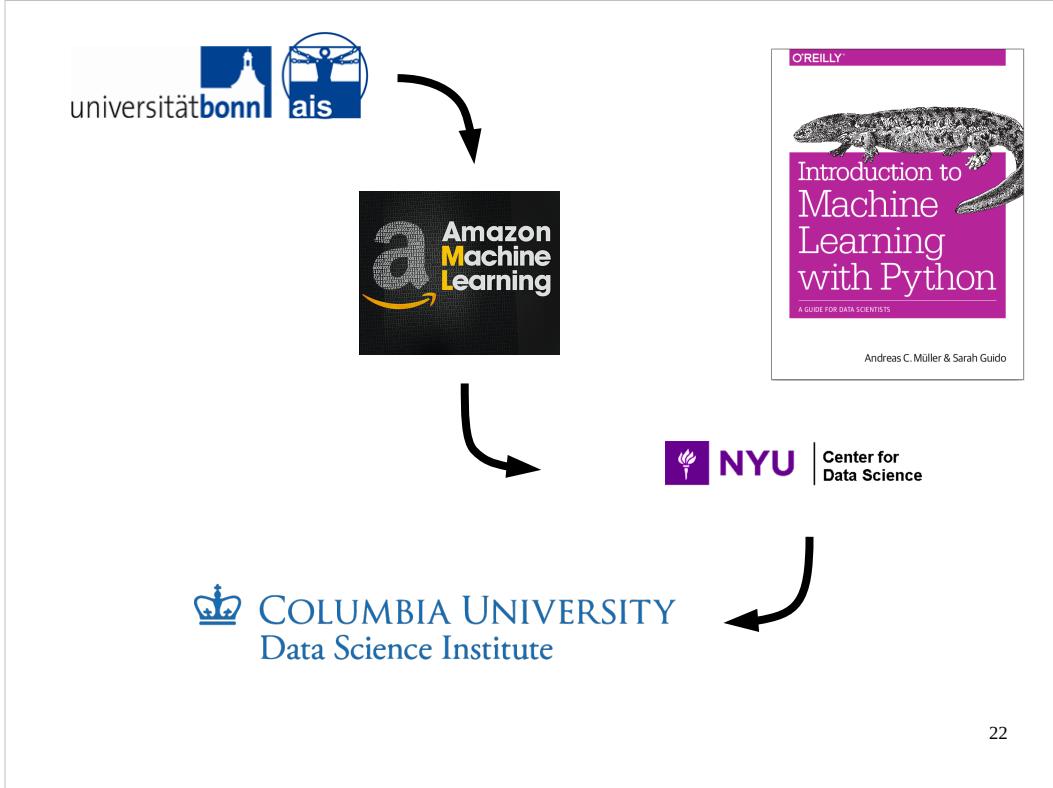


Hi, I'm Andreas Mueller. I'm grateful for the opportunity to talk to you today about my work here at the DSI, and about Open Source for Science and Data Science.

I'm an Associate Research Scientist, and I mostly work on open source software for machine learning, in particular the scikit-learn library.

My talk will have two parts: first, I want to talk a bit about the importance of software for science, and the scientific python ecosystem.

Then I'll talk more about the project I'm working on, scikit-learn, and my group here at Columbia.



22

Very briefly some background on myself. I did my PhD in Machine learning at the University of Bonn in Germany. Then I worked for the Amazon Machine Learning Research lab in Germany in Berlin. During my PhD I had started contributing to open source, and I couldn't really do that well while at Amazon.

I was then lucky to get a position at NYU as a Research Engineer as part of the moore-sloan data science environment, where I continued my open source work and also co-authored this book about machine learning with python.

After two years at NYU I moved to the DSI, first as lecturer and now as Associate Research Scientist.

Why Software Matters

“An article about computational science in a scientific publication is not the scholarship itself, it is merely advertising of the scholarship.

The actual scholarship is the complete software development environment and the complete set of instructions which generated the figures.”

- Buckheit and Donoho (1995)

I care a lot about software, because I think software is an important tool for science.

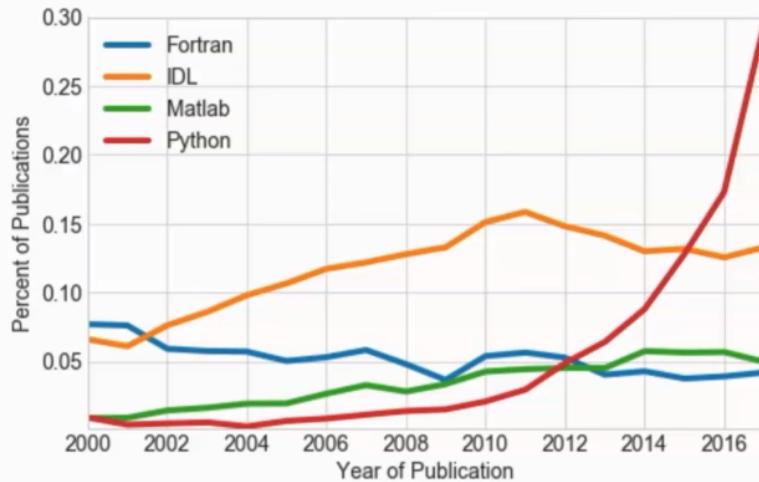
Here is a well-known quote on reproducible research that I really believe in. It says (...)

The actual scholarship is the software, and the data. And so how this software is written is very important to how good the science is. In particular, bad software leads to bad science, while good, and easy to use tools enable good science and reproducible science.

In particular, we want open software, that's available for everybody, and that can be reviewed and inspected. This is why I and others push for the adoption of open source principles in scientific software – and are quite successful.

I see myself very much as a tool-builder that provides tools for others to do science. I don't do a lot of research myself, but the software I build is used by many researchers across disciplines.

Mentions of Software in Astronomy Publications:



Compiled from NASA ADS ([code](#)).

Thanks to Juan Nunez-Iglesias,
Thomas P. Robitaille, and Chris Beaumont.

44

The software I'm working on is part of what's known as PyData or the scientific python ecosystem. This ecosystem has found tremendous adoption in science. Here is an example from Astronomy, where there has been spectacular adoption.

People move away from low-level languages like Fortran and C++ towards python because it is much easier to use and integrates better into a scientific workflow. People also move away from Matlab, because Python is more versatile, and because it's free and open and available to everybody. The tools for science shouldn't be locked away behind some license agreement.

You could plot similar graphs for other disciplines, like biology or geology or neuroscience and would see similar trends.

Engineering is still locked in with matlab a bit, but they are also moving towards python.

The other big player in this field is the R programming language, which is a domain specific language for statistical analysis. Some fields that have more of a history in statistics prefer using R over python, though Python is more broadly applicable.

This is not an either-or. Both are important tools for scientists. R is also open source and has many great tools for statistical analysis and visualization available.

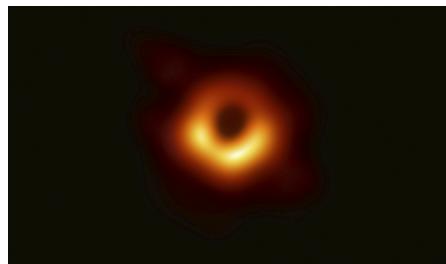
Impact

"The scientific Python ecosystem is critical infrastructure for the research done at LIGO."

David Shoemaker, LIGO Scientific Collaboration

These [SciPy/PyData] are fundamental to training biologists to have the computational skills they need to be effective in today's research realm.

Bonnie Hurwitz, Hurwitz lab, iPlant



55

Here are some more data points on the use of the scientific python ecosystem. It's critical for LIGO, and the discovery of gravitational waves was made using these tools.

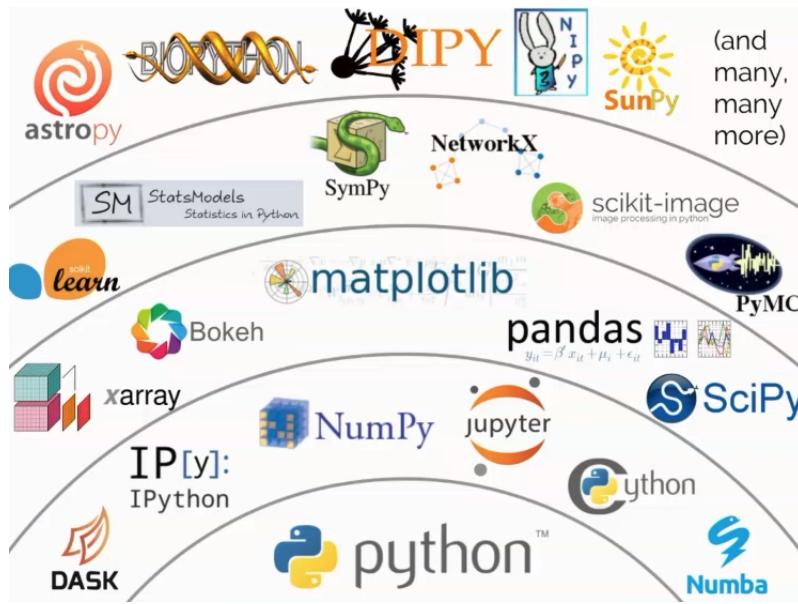
It's indispensable in biology and training biologists.

You might have seen this picture of a black hole recently, it, of course, has also been made using the scientific python ecosystem.

The Scientific Computing Center (NERSC) found that about half of the users of their supercomputer use Python, and that NumPy, SciPy, Matplotlib, Pandas and scikit-learn are the five most frequently used Python libraries among NERSC users.

Basically, you can say, a large fraction of computational science that's done in any discipline, anywhere in the world, builds on the scientific python ecosystem.

The Scientific Python Ecosystem

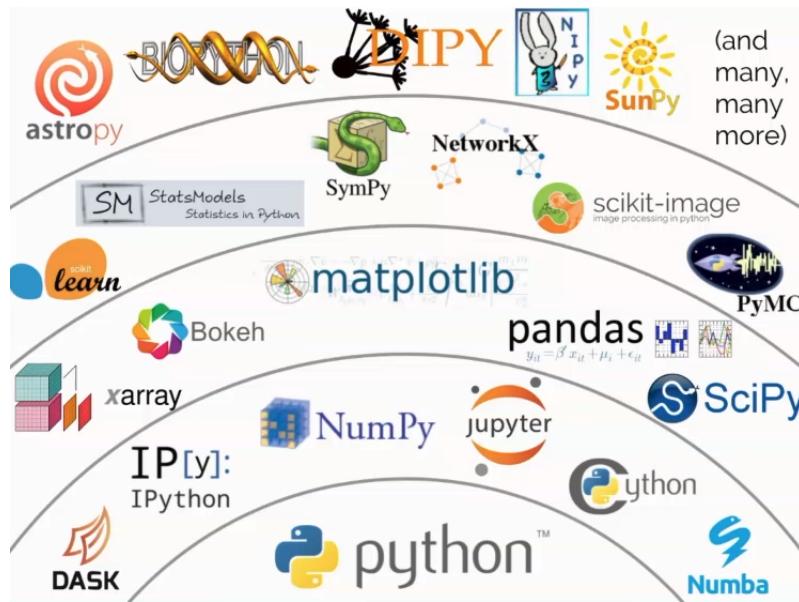


Let me explain a bit more what I mean by the scientific python ecosystem. Sometimes I and others just say “Python”. Python is the programming language that is used to build all these tools. But the real strength of the ecosystem are all the tools that build on top of it.

So at the bottom we have the language Python. Above that we have tools for efficient numeric computation, in particular numpy and cython, and tools for interacting with the computation in an intuitive way, jupyter and ipython. These build on the language, of course.

The layer above builds on these in turn, and consists of libraries for working with n-dimensional data, doing data visualization, doing optimization and doing ETL. Everybody that uses Python in science or industry uses these tools.

The Scientific Python Ecosystem



77

Building on these are more specialized libraries, like scikit-learn for machine learning, which is what I'm primarily working on. There's also statsmodels, and networkx for graph computation, probabilistic modelling, image processing and others, which are specialized but applicable across domains.

Finally on top of the stack, we have domain specific packages like astropy, biopython, dipy and nipy for neuro imaging, sunpy for solar physics and many more.

So when someone says Python in the context of science or data science, what is meant is the collection of all of these projects working together.

So far, that all sounds great, right? And actually both the python community and adoption in the scientific community have made strides and I'm very happy about this. But not all is good.

Transforming Science Through Cyberinfrastructure

NSF's Blueprint for a National Cyberinfrastructure Ecosystem for Science and Engineering in the 21st Century

Executive Summary

Twenty-first century science and engineering (S&E) research is being transformed by the increasing availability and scales of computation and data. The national cyberinfrastructure (CI) ecosystem has thus become a key catalyst for discovery and innovation and now plays a critical role in ensuring US leadership in S&E, economic competitiveness and national security, consistent with NSF's mission. The vision and blueprint presented in this document have been developed by the NSF Office of Advanced Cyberinfrastructure (OAC) on behalf of NSF based on a synthesis of multiple community inputs through advisory bodies, requests for information (RFIs), workshops and conferences, and national initiatives.

A new vision. NSF envisions ***an agile, integrated, robust, trustworthy and sustainable CI ecosystem that drives new thinking and transformative discoveries in all areas of S&E research and education.*** This vision embodies the following overarching principles:

88

Let's look at this roadmap document from the NSF. This is the current draft from April about how to transform science through cyberinfrastructure – which mostly means software.

Transforming Science Through Cyberinfrastructure

*NSF's Blueprint for a National Cyberinfrastructure Ecosystem for
Science and Engineering in the 21st Century*

No mention of Python

Executive Summary

Twenty-first century science and engineering (S&E) research is being transformed by the increasing availability and use of computation and data. The national cyberinfrastructure (CI) ecosystem has thus become a key catalyst for discovery and innovation and now plays a critical role in ensuring US leadership in S&E, economic competitiveness and national security, consistent with NSF's mission. The vision and blueprint presented in this document have been developed by the NSF Office of Advanced Cyberinfrastructure (OAC) on behalf of NSF based on a synthesis of multiple community inputs through advisory bodies, requests for information (RFIs), workshops and conferences, and an initiative.

A new vision. NSF envisions ***an agile, integrated, robust, trustworthy and sustainable CI ecosystem that drives new thinking and transformative discoveries in all areas of S&E research and education.*** This vision embodies the following overarching principles:

99

No mention of R

No mention of SciPy

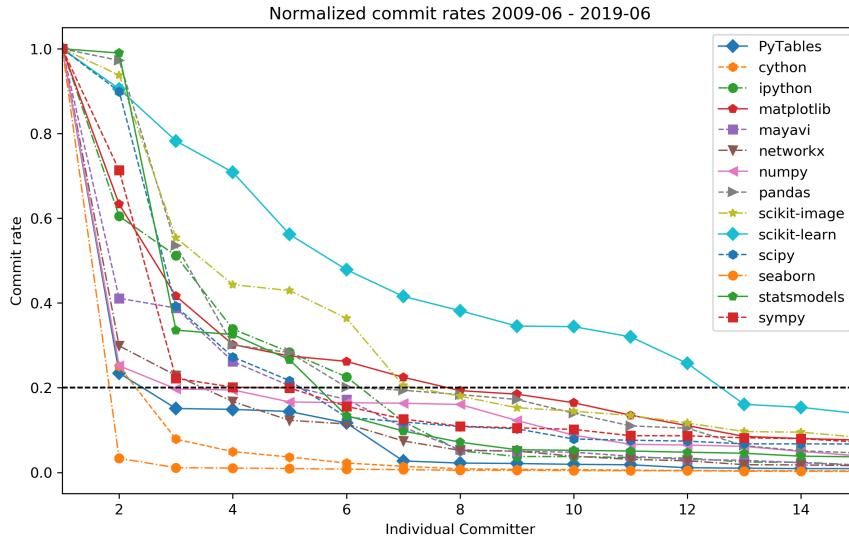
No mention of Open Source

Shockingly, this document doesn't mention Python or the scientific python ecosystem. It also doesn't mention R. In fact, it doesn't mention even the concept of open source.

I just laid out to you that the scientific python stack is the computational basis of most of the science that's funded by the NSF. Yet the vision completely ignores it.

Maybe I have time to come back to that later, but let's leave it at this and talk about scikit-learn now.

Support and Bus Factor



1010

That lack of support has some dire consequences.

Here is a plot of contributions to the major packages of the scipy ecosystem, normalized across contributors. On the right is the most prolific contributor, then the second most prolific and so on. This is aggregated over the last 10 years.

What you can see here is that for many projects, the majority of the work might have been done by only one or two developers. These are mostly people working in their free time out of idealism. If any of these top contributors would leave the project, this would significantly harm the ecosystem. Yet this work is not usually even part of their job, they fit it in on the side.

There is the somewhat macabre notion of bus factor, which says “how many people have to be run over by a bus for this project to die”. In reality, recently it’s more the “google bus” that hires devs away from academia to industry jobs where they have less free time to work on open source.

This issue that’s pervasive in the ecosystem leads to what I call the scholz factor.

Scholz Factor



1111

There is this conference every year in Austin where many of the developers from the scipy ecosystem gather. And there's a social event at this nice place here, Scholz Beer Garden.

We are right now in a situation, where if someone dropped a bomb on this beer garden on the right day in July, basically the whole computational infrastructure for science, globally, would collapse.

I know it sounds like I'm exaggerating. The sad part is, I am not. There are maybe 50 people in the world that in their free time develop and maintain the whole infrastructure that computational science relies on, and most of them don't get anything for it.

I will come back to this later, but now let me switch gears and talk about scikit-learn, the project I'm working on.



Machine Learning with Python and Scikit-learn

1212

Scikit-learn, as I said, is part of the scientific python ecosystem. And it's the main tool to do machine learning within this ecosystem, and arguably the most popular tool to do machine learning overall.

Scikit-learn is a tool mainly for supervised learning, but also unsupervised learning, preprocessing, model selection, model evaluation and other tools around machine learning.

Our Mission:
Commoditize and Democratize Machine Learning

1313

The mission of scikit-learn is to commoditize and democratize machine learning.

We want to enable everybody to use machine learning and data science. We want to make it easy, and we want to make it easy to do it right.

We are not at the cutting edge of research, we are building tools that actually work, and work robustly, and are well documented and easy to use.

And I would say we've been quite successful.

Authors

The following people are currently core contributors to scikit-learn's development and maintenance:










Joris Van den Bossche	Loïc Estève	Thomas J Fan	Alexandre Gramfort	Olivier Grisel	Yaroslav Halchenko	Nicolas Hug
						
Adrin Jalali	Guillaume Lemaître	Jan Hendrik Metzen	Andreas Mueller	Vlad Niculae	Joel Nothman	Hanmin Qin
						
Bertrand Thirion	Tom Dupré la Tour	Nelle Varoquaux	Gael Varoquaux	Roman Yurchak		

Emeritus Core Developers

Alexander Fabisch
Alexandre Pasos
Angel Soler Gollonet
Antoine Ravat
Chris Gohlke
David Cournapeau
David Warde-Farley
Eduard Duchesnay
Fabian Pedregosa
Gilles Louppe
Jacob Schreiber
Jake Vanderplas
Jaques Grobler
Jarrad Millman
Kyle Kastner
Lars Buitinck
Manoj Kumar
Mathieu Blondel
Matthieu Brucher
Noel Dawe
Paolo Losi
Peter Prettenhofer
Raghav Rajagopalan
Robert Layton
Ron Weiss
Sébastien Raschka
Shihao Du
Thomas (Ray) Jones
Vincent Dubourg
Vincent Michel
Virgile Fritsch
Wei Li

1,323 contributors

1414

There is currently 19 core developers, with a long list of people previously engaged. In total the project has had contributions from over 1,300 developers, including researchers at at least 34 US universities.

The project was started around 2010 by a team working on Neuroscience at INRIA in Paris, and that team is still very involved. For example Alex Gramfort, Olivier Grisel, Bertrand Thirion and Gael Varoquaux are among the original authors.

I have been contributing to the project for about 8 years now and I was the release manager for most of that time.

Both I and INRIA have people working on the project on the clock, but a lot of the contributions are done by people in their free time. In particular by Joel Nothman, who has been the most prolific contributor in the last several years.

Authors

The following people are currently core contributors to scikit-learn's development and maintenance:



Joris Van den Bossche



Loïc Estève



Thomas J Fan



Alexandre Gramfort



Olivier Grisel



Yaroslav Halchenko



Nicolas Hug



Adrin Jalali



Guillaume Lemaître



Jan Hendrik Metzen



Andreas Mueller



Vlad Niculae



Joel Nothman



Hanmin Qin



Bertrand Thirion



Tom Dupré la Tour



Nelle Varoquaux



Gael Varoquaux



Roman Yurchak

Emeritus Core Developers

Alexander Fabisch
Alexandre Pasos
Angel Soler Gollonet
Anand Mehta
Chris Gorgolewski
David Cournapeau
David Warde-Farley
Edward Duchesnay
Fabian Pedregosa
Gilles Louppe
Jacob Schreiber
Jake Vanderplas
Jaques Grobler
Jarrad Millman
Kyle Kastner
Lars Buitinck
Manoj Kumar
Mathieu Blondel
Matthieu Brucher
Noel Dawe
Paolo Losi
Peter Prettenhofer
Raghav Rajagopalan
Robert Layton
Ron Weiss
Siddhartha Mishra
Shihao Du
Thomas (Ray) Jones
Vincent Dubourg
Vincent Michel
Virgile Fritsch
Wei Li

 **1,323 contributors**

1515

Here I highlighted my current team, Thomas Fan, Nicolas Hug and myself.

We have nearly 3 FTEs on the project, about the same amount as Paris. This makes us some of the biggest contributors to the project and my team has been making many important improvements.

Nicolas has started last summer and Thomas has started this spring. I'm quite excited for what we'll achieve in the next year, and hope we can continue beyond that.

This makes us actually one of the biggest teams working on open source for data science in the US, and certainly the only group working on such a popular tool.

Scikit-learn: Machine learning in Python 17173 2011
F Pedregosa, G Varoquaux, A Gramfort, V Michel, B Thirion, O Grisel, ...
Journal of machine learning research 12 (Oct), 2825-2830

About 100 citations / day
About 4 citations since this meeting started!

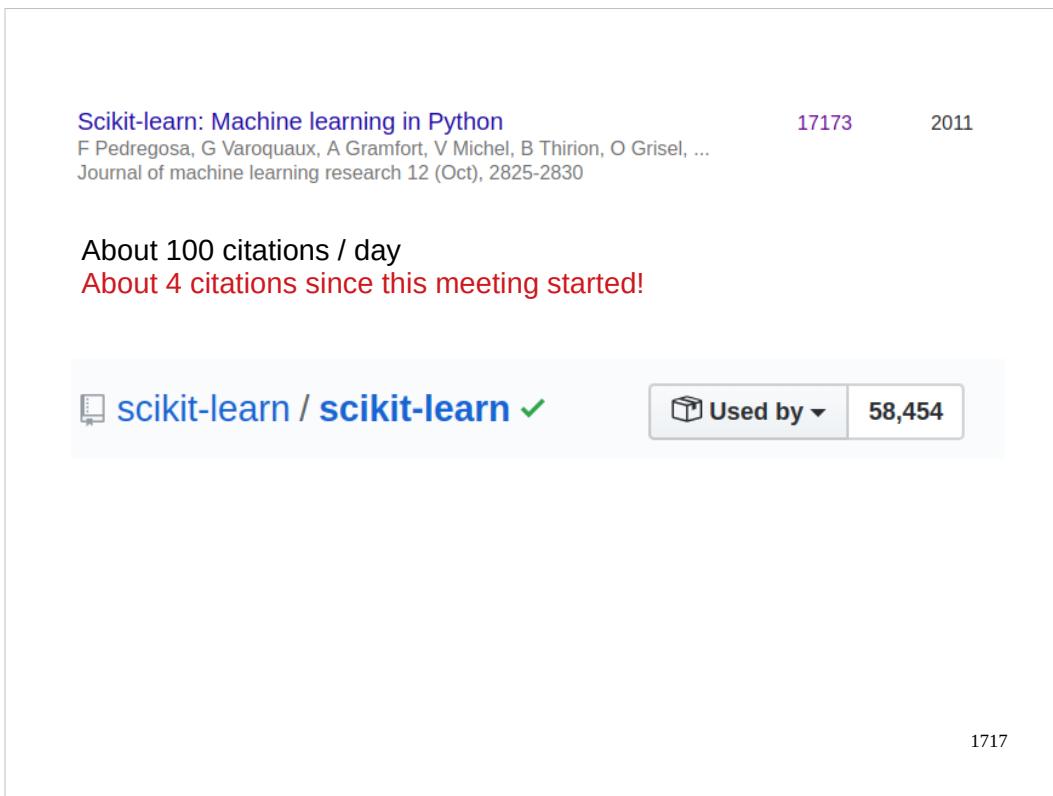
1616

Here are a couple of measures of our success. The paper describing scikit-learn has over 17,000 citations, and it's growing at an amazing rate of about 100 citations a day.

That means since this meeting started, about 4 scientific papers have been published using our software.

Actually, these numbers are likely to underestimate the usage, as software is rarely cited. Personally I think multiplying these numbers by 10 seems more realistic.

So I know 4 people have published research using these tools since the meeting started, but I estimate it was actually 40.



Github tells us that there are nearly 60,000 projects on github that specify scikit-learn as a dependency.

I would say that a large fraction of machine learning projects that are not doing deep learning are driven by scikit-learn. That's true both for academia and industry.

There was a time when the spotify artist radios were done by scikit-learn. Mayor credit card companies use it for fraud detection. Microsoft told me about half of their data scientists use scikit-learn, which is the same amount as use their internal tool.

Scikit-learn: Machine learning in Python
F Pedregosa, G Varoquaux, A Gramfort, V Michel, B Thirion, O Grisel, ...
Journal of machine learning research 12 (Oct), 2825-2830

About 100 citations / day
About 4 citations since this meeting started!

 scikit-learn / scikit-learn ✓  Used by ▾ 58,454

 François Chollet ✅
@fchollet

Replying to @ylecun

The only thing Keras took from Torch7 is the name of the Sequential class. Checking out early Keras versions demonstrates they have virtually nothing in common. The promo line "in the spirit of Torch" was only meant as "it's simple/minimalist". Sklearn was a far bigger influence.

1818

While scikit-learn doesn't do deep learning, and is pretty conservative in its scope, it has inspired other packages that mimic it.

The user interface we created was the blueprint for deep learning packages like keras, which is now developed by google.

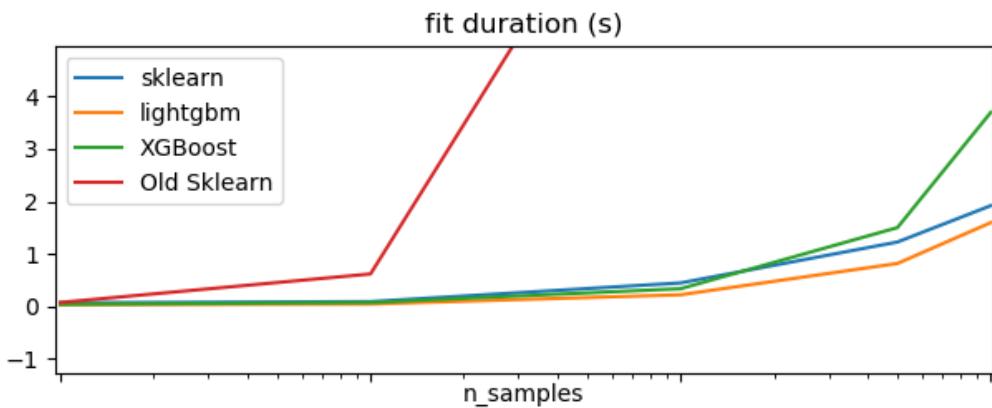
Current Work

1919

Ok, enough bragging, I want to talk a little bit more about what my team is currently working on.

A lot of the work we do is quite technical and I don't want to go into the details here, but I want to give some highlights on the more user-facing side.

Extreme Gradient Boosting

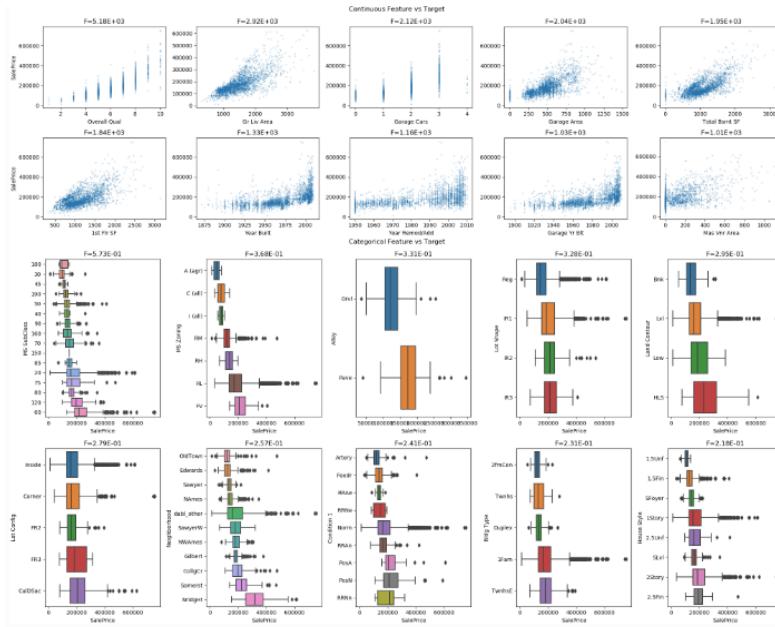


2020

One of the recent additions by my team is an implementation of extreme gradient boosting by Nicolas Hug. This is much, much faster than what was in scikit-learn before and it will have a huge impact on the ML community.

You can see here that it's comparable to two existing tools, xgboost and lightgbm. LightGBM was released by microsoft and it's quite great, but by including this implementation into scikit-learn, we make it available to a much wider audience.

Visualization

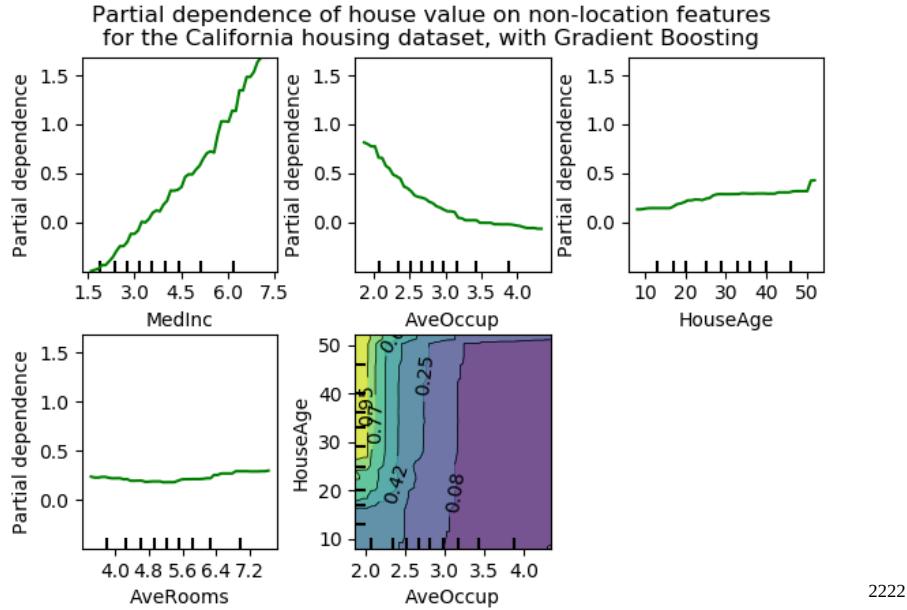


2121

One of the other areas that we're working on is visualization. We're adding new visualization tools to scikit-learn, and we're also building new tools that automatically create relevant and informative visualizations for machine learning tasks.

I think the human in the loop in data science is very important and I want to make it as easy as possible for the user to look at the data in different ways.

Model understanding and inspection



Related to that is model understanding and inspection. We're building better tools that allow users to validate and debug their models, understand their data better, whether they should trust their model, and what they can learn from it.

This allows our users to better judge when and why their model might work or fail.

AutoML & dabl

```
>>> types = dabl.detect_types(titanic_clean)
>>> print(types)
      continuous  dirty_float  ...  free_string  useless
pclass          False        False  ...           False  False
survived        False        False  ...           False  False
name            False        False  ...            True  False
sex             False        False  ...           False  False
sibsp           False        False  ...           False  False
parch           False        False  ...           False  False
ticket          False        False  ...            True  False
cabin           False        False  ...            True  False
embarked        False        False  ...           False  False
boat            False        False  ...           False  False
home.dest       False        False  ...            True  False
age_?           False        False  ...           False  False
age_dabl_continuous  True        False  ...           False  False
fare_?           False        False  ...           False  True
fare_dabl_continuous  True        False  ...           False  False
body_?           False        False  ...           False  False
body_dabl_continuous  True        False  ...           False  False
[17 rows x 7 columns]
```

```
sc = SimpleClassifier()
sc.fit(data, target_col="survived")
```

```
ac = AnyClassifier()
ac.fit(data, target_col="survived")
```

2323

Finally, we're working on tools build on top of scikit-learn for more automated machine learning.

What we're trying to automate is parameter tuning and selecting the right algorithms. These are things that are actually fairly easy to automate to some degree.

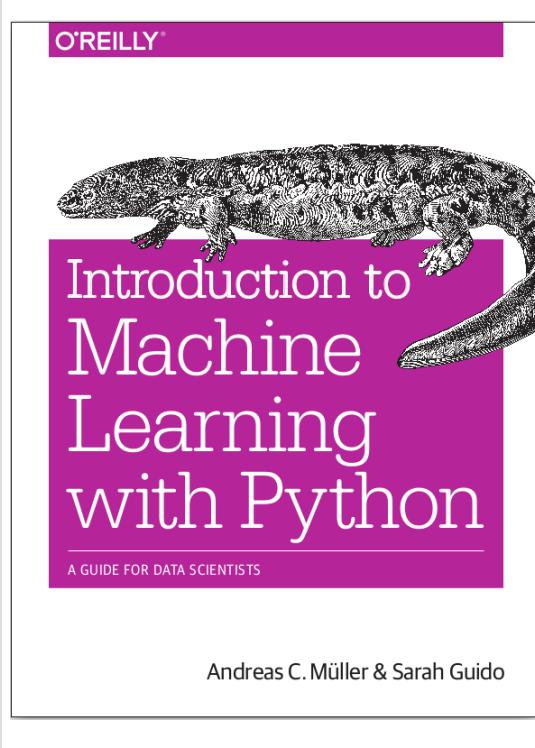
The hard part in machine learning is collecting and preparing the data and understanding the interaction between data and model. So we're automating one part, but the goal is to enable the user to iterate more quickly on other aspects. I want the human in the loop, but I don't want the user to waste their time tuning parameter. I want them to interpret the results. So dabl tries to create an interactive loop in which the user can spend their time where it matters.

Where to go from here

2424

So I think there is a lot of very exciting and impactful work that we can do here, and reduce the barrier to entry for machine learning even more, both for academia and industry.

Ideally I would like to see more people working on building tools. Investing in this infrastructure is hugely productive, both for the scientific community and industry, and the contributions my team could make have already been tremendously useful.



amueller.github.io



@amuellerm!



@amueller



andreas.mueller
@columbia.com

Thank you for your attention and I'm happy to take any questions!