ORACLE
Data Cloud

Open Source Tools for Enterprise Data Science

An analysis of the open source trends driving change in the data science space.

WHITE PAPER / UPDATED MAY 2019

ORACLE

INTRODUCTION

The field of data science has been steadily gaining a foothold in the corporate sector over the past decade. It is now an integral part of business strategy for some of the world's most successful companies. As the scope of enterprise data science changes, so too have the tools data scientists are using to solve complex problems, from building models for identifying and retaining high-value customers to creating highly effective product recommendation engines.

This white paper evaluates the following open source players in the data science space:

- Google's TensorFlow and deep learning library Keras
- Visualization libraries matplotlib and ggplot
- Permissive open source licenses

Proprietary data science solutions—once the mainstay of enterprise data science—are being eclipsed by open source projects like R, Spark, and TensorFlow; in fact, 62 percent of analytics professionals now prefer R or Python to a legacy, proprietary SAS solution.[1] While there are many reasons for this shift, a major one is that open source tools are available to anyone, bringing endless opportunities for collaboration and contribution. Open source tools are no longer considered unreliable or limited; instead, they have been embraced by the data science community at large and built out to the point that they provide measurable value, even in an enterprise capacity.

Instead of fighting its explosive growth, database and data science software providers are jumping on the open source bandwagon. Case in point: Oracle Cloud Infrastructure Data Science, Oracle's enterprise data science platform, enables data science teams of any size to work in Python-based notebooks, Apache Spark, R, and more—using languages and machine learning libraries of their choice.

This white paper leverages data from the GitHub Archive, made public last year through Google's BigQuery, to evaluate a few major open source players in the data science space: Google's TensorFlow and deep learning library Keras, visualization libraries matplotlib and ggplot, and permissive open source licenses.

WHY OPEN SOURCE TOOLS ARE THE KEY TO BEATING YOUR COMPETITION

For companies of any size, open source software adoption brings its own set of challenges. From licensing your own modified versions of open source tools, to creating an appropriately sized open source technology stack, there is no single way to integrate open source solutions into your existing data science workflow. But it's highly advantageous if you do so effectively.

It's easy to miss the mark: In many cases, tool sprawl—working across too many disjointed tools, the leading business problem for data-driven companies[2]—can cripple your team's ability to deliver value. Too few open source tools might mean that you're leaning heavily on expensive, proprietary options to fill the gaps. But just the right amount can spell value for your organization.

Table 1. Actions for GitHub Users

ACTION	DEFINITION
Commits	A commit is a change to a file or set of files. Each commit in GitHub creates a unique ID, providing a record of how many different contributors are iterating on a project in a given time period.
Stars	GitHub users who star a repository are essentially bookmarking it and, in effect, showing appreciation to the creator of the repository for their work. These users aren't necessarily contributing to project.
Pull requests	A pull request is a method of submitting a contribution to open source project.

The analysis examined the frequency of these three types of actions GitHub users can take—commits, stars, and pull requests—to determine the popularity of each open source tool.

DEEP LEARNING

Google's TensorFlow Changes the Hierarchy of Deep Learning Libraries

In November 2015, Google open sourced its software library for machine learning, TensorFlow, kicking off a chain reaction across the deep learning space. While it's no surprise that TensorFlow has been wildly popular—the repository was starred 10,893 times within five days of its initial release—what was less expected was the ripple effect it had on other compatible tools.

One of those tools is Keras, a deep learning framework built by Google software engineer and artificial intelligence researcher François Chollet in March 2015 to provide "a set of Lego blocks for building deep learning models in a fast and straightforward way."[3] What Keras doesn't do is handle low-level tensor operations, so it has to sit on top of another solution. Initially, that solution was Theano, a Python library developed by a machine learning group at the Université de Montréal—but the release of TensorFlow changed all that.

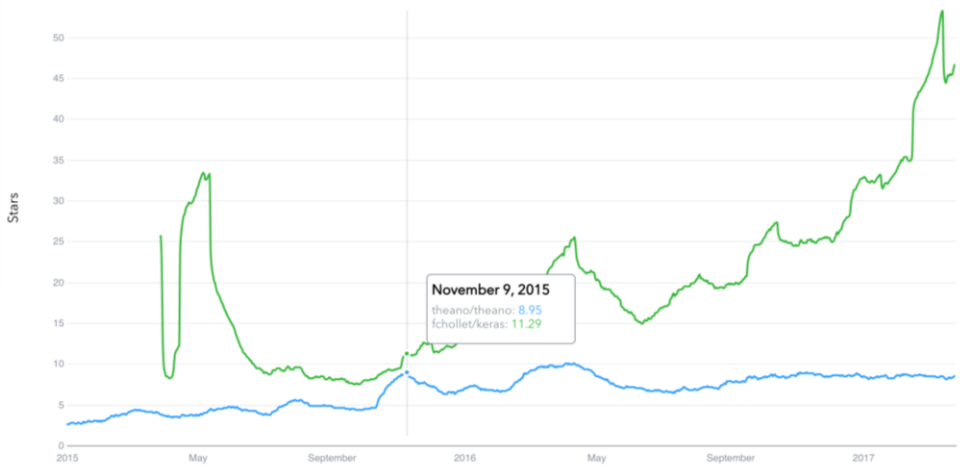


Figure 1. The popularity of Keras eclipses Theano with the release of TensorFlow.

"When we started Keras in March 2015, Theano was the natural choice . . . Since then, there has been a lot of innovation in the symbolic tensor computation space—a lot of it in the footsteps of Theano. Most notably, we've seen two new frameworks appear, Neon from Nervana Systems and TensorFlow from Google. While Neon is the faster framework right now, TensorFlow has the engineering weight of Google behind it and there is no doubt that it will improve considerably over the next few months."

François Chollet
Software engineer and AI researcher at Google, writing in a December 2015 blog post announcing the creation of a TensorFlow backend for Keras

As seen in the chart above, Keras saw a major spike in interest just after its release. But with the introduction of TensorFlow approximately 10 months later, the number of stars has trended steadily upwards, significantly overtaking Theano in mid-2016. In fact, since the release of TensorFlow up until the end of March 2017, Keras has been starred 57,201 times; Theano, just 10,305.

Furthermore, TensorFlow is now being used in more than 8,000 open source repositories (an increase of nearly 2,000 repos since February 2017)[4], considerably outpacing usage of Theano, which is currently being leveraged in approximately 1,500 repositories. That growth is mirrored in adoption of Keras, which now has more than 100,000 users.[5]

The rise of TensorFlow has irrevocably changed deep learning on an enterprise level, with companies such as Snapchat, eBay, Airbnb, and Dropbox all building projects using Google's robust machine learning library. TensorFlow will likely continue to grow its capabilities moving forward, making adoption a good move for companies working on machine learning projects. Its latest release improves the speed and flexibility of model building, in part through the introduction of a new module that is fully compatible with Keras.

"Keras has enabled new startups, made researchers more productive, simplified the workflows of engineers at large companies, and opened up deep learning to thousands of people with no prior machine learning experience."

François Chollet
Software engineer and AI researcher at Google, writing in a March 2017 blog post

Why should companies consider pairing Keras with TensorFlow?

Keras ultimately makes adoption of TensorFlow easier by abstracting away many of the computations needed for creating neural networks, models used in applications such as facial recognition and spam filtering. In addition, its capabilities are making it easier for enterprise companies to work on complex machine learning projects.

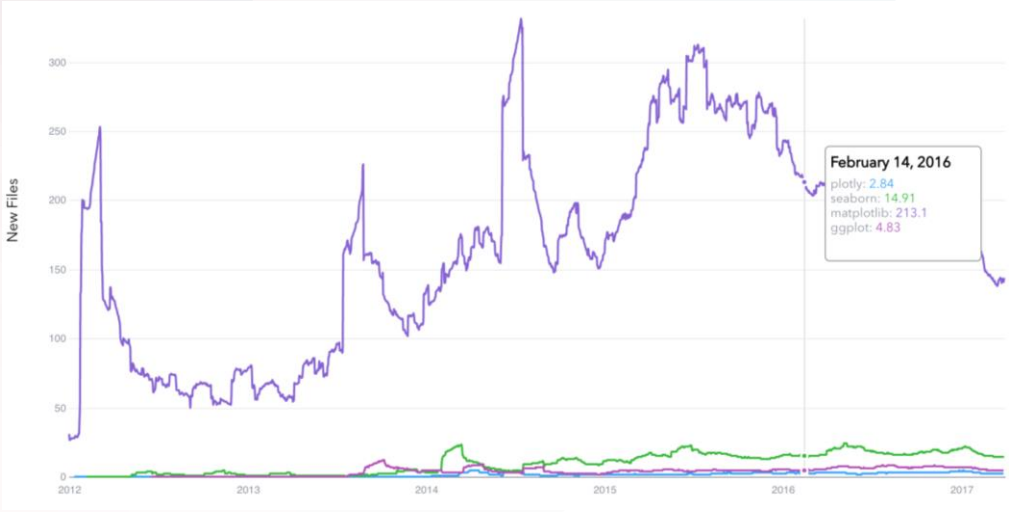
DATA VISUALIZATION

Ggplot Gains Ground Against Data Visualization Giant Matplotlib

Data visualizations are meant to bring clarity to an analysis, making it easier for decision makers to identify trends or patterns in complex datasets. However, visualizing data can quickly become costly—either because a company has opted to use an expensive subscription service like Tableau, or in terms of resources if different teams are needlessly using different visualization tools that require different skillsets.

Standardizing data visualization across your company with an open source tool is both efficient and cost effective. But choosing the right one is both a question of current functionality and the possibility of future development. For some time, the clear winner in this category has been matplotlib, a plotting library written in Python that was first released in 2003.

Standardizing data visualization across your company with an open source tool is both efficient and cost effective. To choose the right one, consider both current functionality and the possibility of future development.



While matplotlib has the highest total number of new files, its momentum is actually slowing down compared to its competitors.

Figure 2. Matplotlib consistently dominates data visualization space.

In the chart above, you can clearly see that matplotlib dominates the open source data visualization space. Matplotlib is consistently more popular than Seaborn, Plotly, and ggplot based on the aggregated number of new files committed per day that mention each library, dating back to 2012. For the purposes of this analysis, file commits were attributed to a certain library if there was a mention of the library in the raw text contents of the file.

Matplotlib is both versatile and clunky; the library itself establishes very few design decisions by default, ultimately leaving it up to the user to spend extra time establishing how he or she wants certain plots to look. Other tools have tried to improve upon its features, including a few in this analysis: Seaborn was built on top of matplotlib and includes built-in themes that seek to enhance standard matplotlib plots, but Seaborn users often find themselves reverting to matplotlib commands while fine-tuning.

Similarly, ggplot was built on top of matplotlib when it was ported into Python. Originally written in R as ggplot2, ggplot enables the user to programmatically define a graph by concatenating high-level visualization components together, rather than requiring the user to repetitively specify low-level features such as axis ticks and marker sizes. But while ggplot's syntax is powerful, but it can be daunting to the inexperienced user. As a result, ggplot tends to have a steeper learning curve than other data visualization libraries.

Outside of the matplotlib realm is Plotly, an active innovator in the data visualization world in recent years. In addition to a plotting library built on top of d3.js and stack.gl (devoid of any matplotlib dependencies), Plotly's enterprise version boasts rich and interactive graphs, automated cloud storage, dashboard capabilities, and APIs spanning Python, R, JavaScript, and beyond. Users of the open source version also have access to APIs that connect it to common data science languages like Python and R.

Despite Plotly's robust offerings, and the supposed improvements to matplotlib offered by Seaborn and ggplot, it would seem that matplotlib will continue to dominate the data visualization space. But the absolute number of files created per day that can be attributed to matplotlib only tells part of the story.

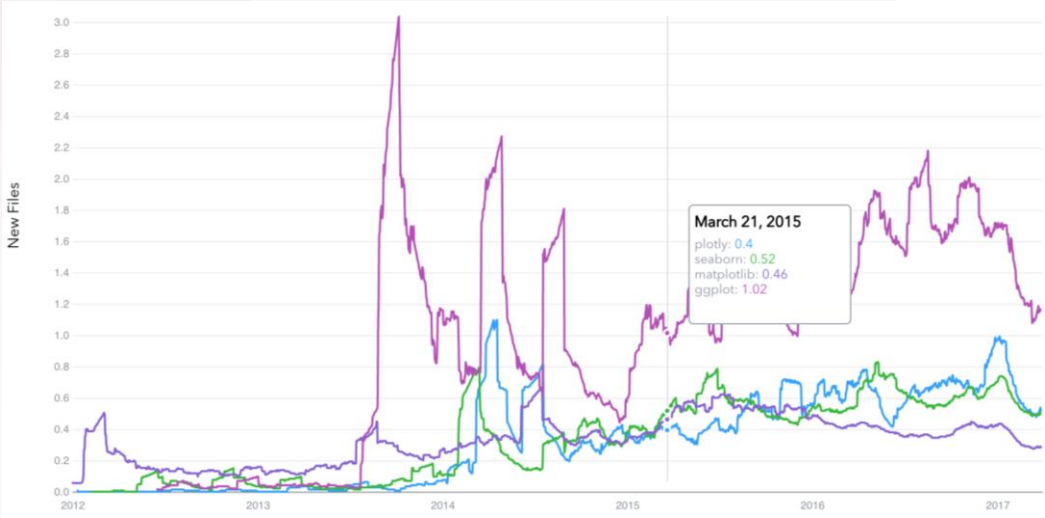


Figure 3. Matplotlib's momentum slows while Ggplot contributions pick up.

The normalized figure above, in which the rate that new files are being added to each library is shown relative to the library's average rate, clearly demonstrates that matplotlib's momentum is actually slowing down compared to its competitors.

Most strikingly, ggplot's adoption has been surging since mid-2013. At that time, Python users could only access ggplot via matplotlib's style sheet workaround.[6] Then in September 2014, Yhat released its first attempt at porting the popular R plotting library over to the Python community.[7] While the initial release was met with limited success, Yhat corrected several issues when it revamped the port in 2016.[8]

Plotting the frequency of new Python files on GitHub containing "ggplot" brings this timeline to life: the introduction of the matplotlib style sheet in 2013, the initial spike in popularity and subsequent disappointment with Yhat's first port in 2014, and the final sustained success of Yhat's revised port in 2016. This emerging trend indicates that ggplot's high-level visualization grammar is gaining acceptance in the open-source community and changing the way practitioners approach data visualization.

SOFTWARE LICENSING

MIT Remains License of Choice for Open Source Projects

From Airbnb's Airflow, an open source workflow management platform built in Python, to Stitch Fix's public collection of projects written in Ruby, Python, and JavaScript, the open source space is seeing an influx of contributions from both major and up-and-coming companies. In fact, 67 percent of companies actively encourage developers to contribute to open source projects.[9]

Companies choose to open source for a number of reasons: recruitment, media exposure, and improvement through crowd sourcing. But whatever the reason, it's imperative that they license open source projects appropriately to prevent legal issues or bar would-be competitors from selling modified versions. Google's more than 2,000 open source projects only use code under certain licenses and the company requires contributor license agreements for all the patches it receives.[10]

Reasons companies choose open source tools include:

- Recruitment
- Media exposure
- Improvement through crowd sourcing

There is a wide array of open source licensing options, but some of the most popular are Apache, MIT, and the GNU General Public License (GPL). Apache and MIT are on one side of the spectrum—both permit anyone to use your code, while offering some level of protection for you as the creator. Apache offers a patent license and a retaliation clause, while MIT simply waives your liability. GPL is a *copyleft* license, meaning any improvements made to your code by outside parties will need to be open sourced as well.

The increased protections in the Apache license have made it attractive to companies like Airbnb, which opted to open source Airflow under the license.[11] The Apache license has also successfully garnered recognition for software developers like the creators of etherpad, an open source online editor that became the basis of Hackpad. Hackpad was acquired by Dropbox in 2014, and its code was open sourced under the Apache license—essentially, code from etherpad is now integral to the workflow of every user on Dropbox Paper.

Increased protections in the Apache license have made it attractive to companies, yet the MIT license is still the dominant license due to its permissiveness and simplicity. However, both these licenses are losing share; the GPLv3 license is the only one trending upward.

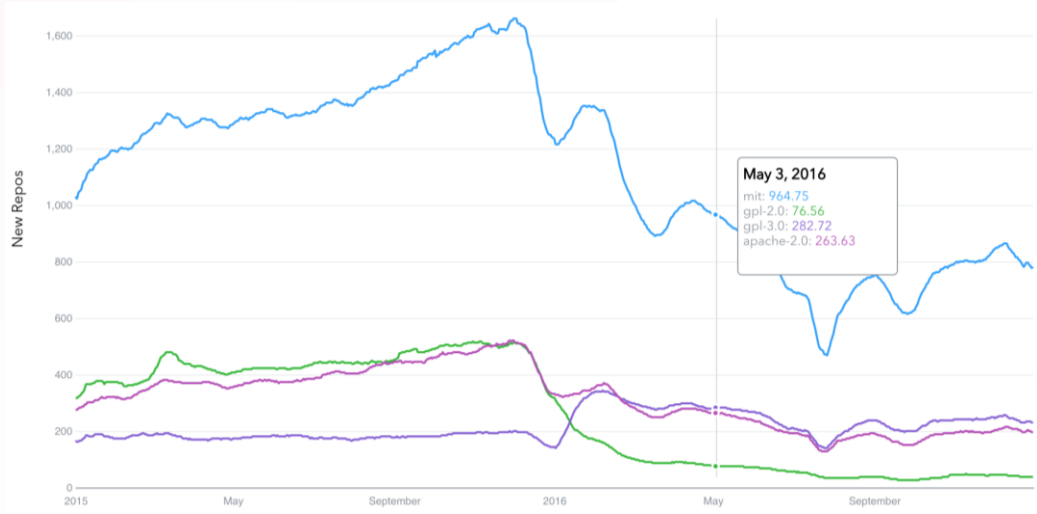


Figure 4. MIT is the license of choice for protecting open source projects.

Even so, as seen in Figure 4 above, the MIT license is far and away the dominant license—no doubt owing to its permissiveness and simplicity. GitHub also came to this conclusion in 2015[12] when it found that 44.69 percent of licensed projects in its archive used MIT. To update that finding, we queried the number of new public GitHub repositories created per day for each license type since then.

However, it appears that the popularity of the MIT and Apache licenses is beginning to wane. Instead, GPLv3, the third version of the copyleft license published in 2007[13], is the only license on the chart trending upward.

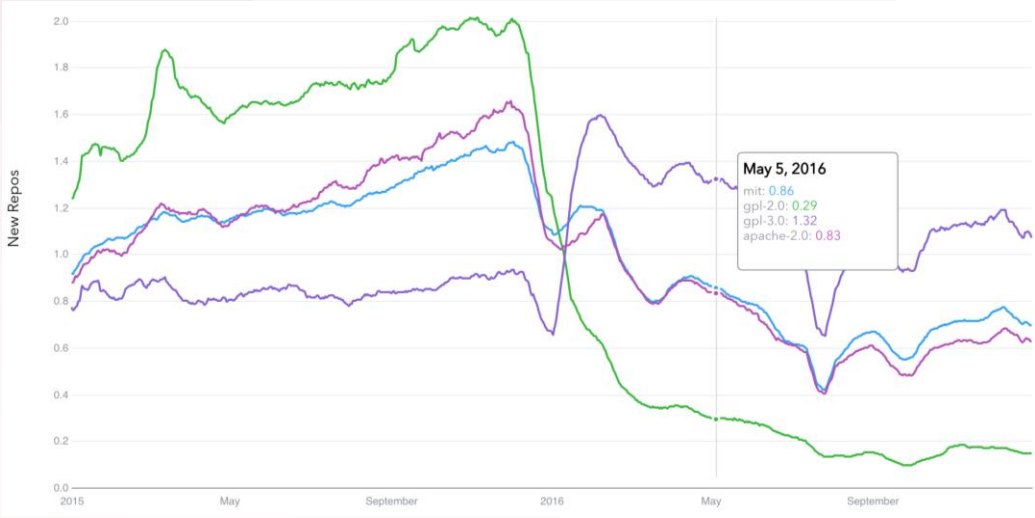


Figure 5. 'Copyleft' GPL license sees growth in 2016.

To investigate this trend, Figure 5 normalizes each time series by dividing by its mean value, revealing an interesting reversal: The first quarter of 2016 saw a 60 percent spike in new repos using the GPLv3 license (relative to its mean), eventually setting around a 10 percent bump above its mean by the end of 2016. On the other hand, MIT and Apache popularity depreciated about 20 to 40 percent by the end of 2016.

GPL's strong copyleft policy can be a turnoff for corporations that seek to profit from code based on GPL-licensed projects, but GPL can also foster community amongst open-source enthusiasts. Although corporate wariness of GPL licenses has discouraged GPL-licensed authorship in the past, the trend highlighted above may represent the first inklings of GPL's future mainstream adoption across the open-source community and at an enterprise level.

THE FUTURE IS OPEN SOURCE

Open source software is steadily gaining support across every industry, with established companies such as Facebook, Microsoft, and General Electric pouring money and resources into public-facing projects. As corporations increasingly rely on data science to get value from their big data, so too will they embrace the open source tools that primarily make up the artificial intelligence, Internet of Things, and data infrastructure space.

The challenge is identifying which of those tools is relevant and valuable to your business. GitHub is adding millions of projects every year; in fact, while the first million repositories were created in just under four years, the million added by the end of 2013 took just 48 days.[14] Assessing the maturity of these projects, grappling with any licensing issues, and making sure your team has the correct skillset to use them are challenges that many companies are now facing.

Understanding the trends in open source software contribution and usage will go a long way in creating a tech stack that makes sense for the data scientists at your organization.

REFERENCES

1. Burtch Works, “SAS, R, or Python Survey 2016: Which Tool Do Analytics Pros Prefer?,” July 13, 2016. burtchworks.com/2016/07/13/sas-r-python-survey-2016-tool-analytics-pros-prefer/

2. Forrester Consulting, “Data Science Platforms Help Companies Turn Data Into Business Value,” December 2016.

3. Francois Chollet, “Keras, now running on TensorFlow,” December 1, 2015. blog.keras.io/keras-now-running-ontensorflow.html

4. Francois Chollet, “Keras, now running on TensorFlow,” December 1, 2015. blog.keras.io/keras-now-running-ontensorflow.html

5. Francois Chollet, “Introducing Keras 2,” March 14, 2017. blog.keras.io/introducing-keras-2.html

6. “Customizing plots with style sheets,” matplotlib.org/users/style_sheets.html

7. “Ggplot for Python,” pypi.python.org/pypi/ggplot

8. The Yhat Blog. “A new ggplot is here,” blog.yhat.com/posts/new-ggplot.html

9. Haidee LeClair, “65% of companies are contributing to open source projects,” opensource.com/business/16/5/2016-future-open-source-survey

10. “Noto Serif CJK is here!,” opensource.googleblog.com/

11. Maxime Beauchemin, “Airflow: a workflow management platform,” June 1, 2015. nerds.airbnb.com/airflow/

12. Ben Balter, “Open source license usage on GitHub.com,” The GitHub Blog, March 9, 2015. github.com/blog/1964-license-usageon-github-com

13. “GNU General Public License,” June 29, 2007. gnu.org/licenses/gpl-3.0.html

14. Brian Doll, “10 Million Repositories,” The GitHub Blog, December 23, 2013. github.com/blog/1724-10-million-repositories

ORACLE CORPORATION

Worldwide Headquarters
500 Oracle Parkway, Redwood Shores, CA 94065 USA

Worldwide Inquiries
TELE + 1.650.506.7000 + 1.800.ORACLE1
FAX + 1.650.506.7200
oracle.com

CONNECT WITH US

Call +1.800.ORACLE1 or visit oracle.com. Outside North America, find your local office at oracle.com/contact.

datascience.com/blogs facebook.com/oracle twitter.com/oracle

Integrated Cloud Applications & Platform Services

White Paper / Open Source Tools for Enterprise Data Science
Updated June 2019