

# julia IS THE NEW FAVORITE!

## MOVE ASIDE PYTHON & R

The Data science landscape in the last 5 years has evolved around developing new learning algorithms and using them to solve complex problems from machine learning to computer vision. The future will demand us to build models that are computationally intensive with large datasets & numerous variables. 'Time to decisions' will become an important KPI.

In this issue, we will explore Julia, the programming language which shows a huge potential to impact this transition and is very well on the way to becoming the data scientist's favorite.



### WHAT IS SO SPECIAL ABOUT JULIA?

Let's hear from the founders:

*"We want something as usable for general programming as Python, as easy for statistics as R, as natural for string processing as Perl, as powerful for linear algebra as Matlab, as good at gluing programs together as the shell. We want it interactive and we want it compiled. We want it to be as fast as C!"*

That is so fascinating, isn't it? Never was a programming language tailor-made for data science! Hybrid, fast and yet, so simple to learn!



### JULIA'S SUPERPOWERS: LET'S DIVE DEEPER!

- 1 Loads and processes data quicker (10x faster than Python and R)
- 2 Best in class package manager with 3500 packages and functionality to call packages from other programming languages
- 3 Multiple dispatch enables right algorithm for right instance
- 4 Ready to be deployed Machine learning frameworks



### APPLICATIONS FOR JULIA

Julia boasts of a wide variety of applications across industries – be it powering autonomous vehicles or applications across domains such as augmented reality, machine learning and precision medicine.

For the analyst community, recommendation engines are going to get a makeover! Here is a case study snippet of Factorization for inspiration.

PARAMETERES / DATASETS	SIZE (NO.OF INTERACTIONS)	FACTORIZATION TIME ( IN SECS )
Movielens	20 Million	119
Last.fm	0.5 Billion	2913

Performance of Image Classifiers will improve – breaking away from conventional recommendations that Tensor Flow and Pytorch projects should be compiled in C++.



### WHO IS USING JULIA?

The list is massive & growing!



In a nutshell, if you need your codes to run faster, while at the same time taking advantage of the capabilities of a high level language, then Julia is an obvious choice.

*"Why are macroeconomists like myself so interested in and excited by Julia? Because our models are complicated. It's easy to write the problem down, but it's hard to solve it – especially if our model is high dimension. That's why we need Julia."*

- Nobel Laureate Thomas J. Sargent

SOURCES: [Juliacomputing.com](https://juliacomputing.com), [intel.com](https://intel.com) presentation on Julia, Github – Pluto.jl, MIT.edu

CONTACT US TO LEARN ABOUT ECLERX'S ANALYTICS PRACTICE!

eClerx