Democratizing AI with sparklyr

@javierluraschi
RStudio PBC - OSS

RStudio AI

RStudio is a Public Benefit Corporation supporting R. The RStudio AI team is focused on making AI accessible to the Data Science community.

RStudio Server Pro

Take control of your R and Python code

An integrated development environment for R and Python, with a console, syntax-highlighting editor that supports direct code execution, and tools for plotting, history, debugging and workspace management.

RStudio Connect

Connect data scientists with decision makers

Publish and distribute data products across your organization. One button deployment of Shiny applications, R Markdown reports, Jupyter Notebooks, and more.

RStudio Package Manager

Control and distribute packages

Control, organize, and govern your use of R packages to increase reproducibility and decrease the time you spend installing and troubleshooting.

R consortium

The R Consortium is a collaboration between the R Foundation, RStudio, Microsoft, TIBCO, Google, Oracle, HP

FOAS

RStudio and Shiny are affiliated projects of the Foundation for Open Access Statistics. FOAS works to

NUMFOCUS

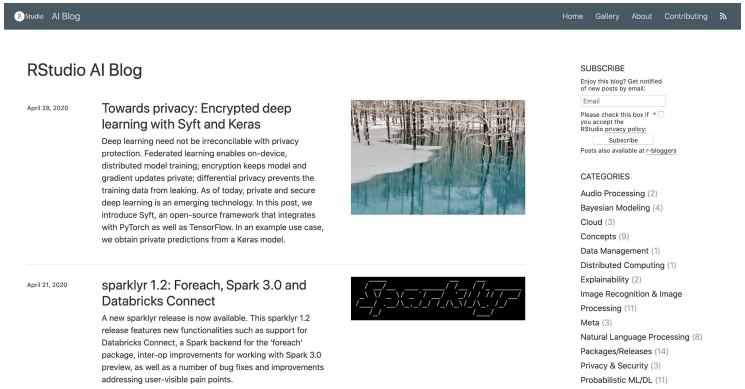
OPEN CODE + BETTER SCIENCE

RStudio is proud to be a corporate sponsor of NumFOCUS, whose mission is to promote open practices

LINUX FOUNDATION

RStudio PBC is a sponsor of the Linux Foundation, which supports the creation of sustainable open source

Walmart, Carnival, Janssen, Stitch Fix, Farmacias del Ahorro, Accenture, AstraZeneca, NBC, GE, The Family Connection Foundation, Honda, The Power of Dreams, Metrum Research Group, Waze, Express Scripts, NASA, Andhia, ebay, DTCC, Liberty Mutual, Santander, Avon, Samsung, Amgen

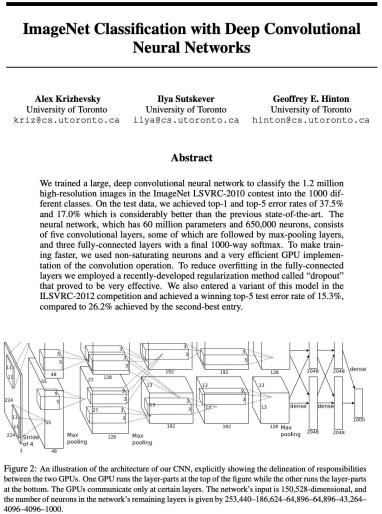


Apache Spark, TensorFlow, PyTorch, mlflow, Horovod

Credit: rstudio.com, blogs.rstudio.com/ai

Artificial Intelligence

Artificial Intelligence is a rapid-growing field with significant transformational impact to many industries and civilization at large.



ImageNet Classification with Deep Convolutional Neural Networks

Alex Krizhevsky
University of Toronto
kriz@cs.utoronto.ca

Ilya Sutskever
University of Toronto
ilya@cs.utoronto.ca

Geoffrey E. Hinton
University of Toronto
hinton@cs.utoronto.ca

Abstract

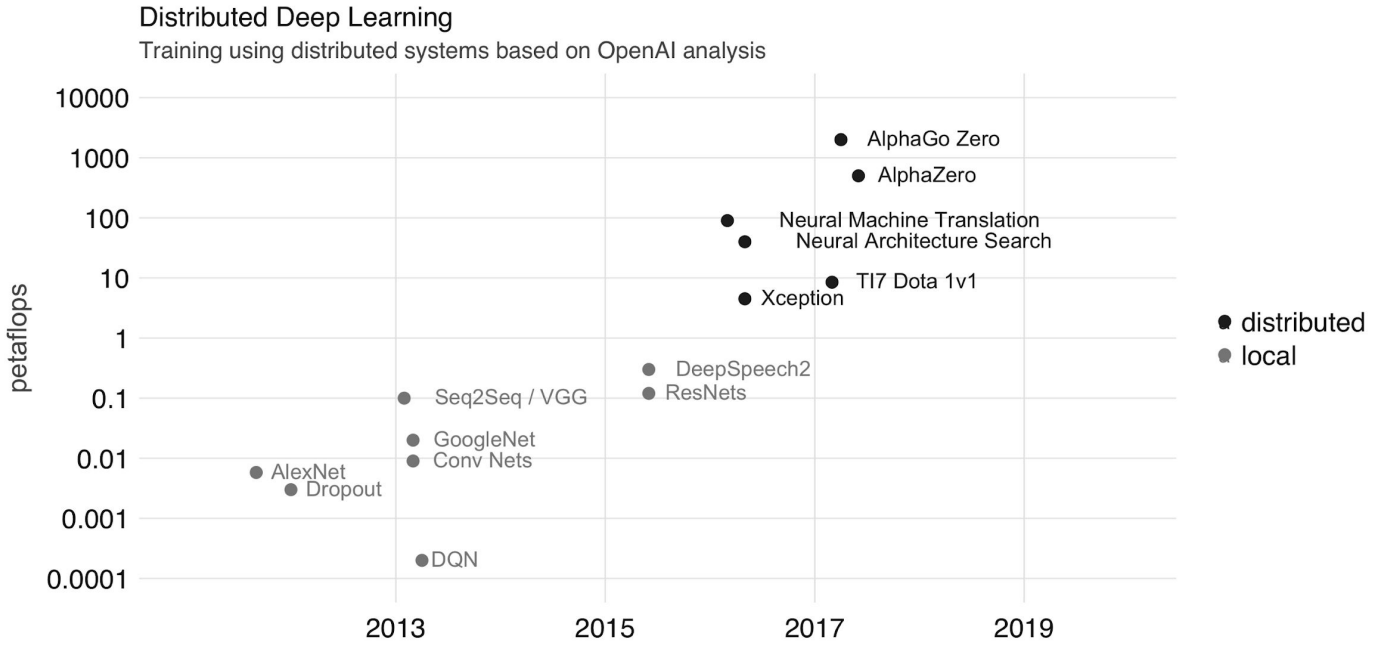
We trained a large, deep convolutional neural network to classify the 1.2 million high-resolution images in the ImageNet LSVRC-2010 contest into the 1000 different classes. On the test data, we achieved top-1 and top-5 error rates of 37.5% and 17.0% which is considerably better than the previous state-of-the-art. The neural network, which has 60 million parameters and 650,000 neurons, consists of five convolutional layers, some of which are followed by max-pooling layers, and three fully-connected layers with a final 1000-way softmax. To make training faster, we used non-saturating neurons and a very efficient GPU implementation of the convolution operation. To reduce overfitting in the fully-connected layers we employed a recently-developed regularization method called "dropout" that proved to be very effective. We also entered a variant of this model in the ILSVRC-2012 competition and achieved a winning top-5 test error rate of 15.3%, compared to 26.2% achieved by the second-best entry.

Figure 2: An illustration of the architecture of our CNN, explicitly showing the delineation of responsibilities between the two GPUs. One GPU runs the layer-parts at the top of the figure while the other runs the layer-parts at the bottom. The GPUs communicate only at certain layers. The network's input is 150,528-dimensional, and the number of neurons in the network's remaining layers is given by 253,440–186,624–64,896–64,896–43,264–4096–4096–1000.

Modern AI starts with the deep learning breakthrough from AlexNet and more recently with achievements like DeepMind's AlphaGo.

Artificial Intelligence

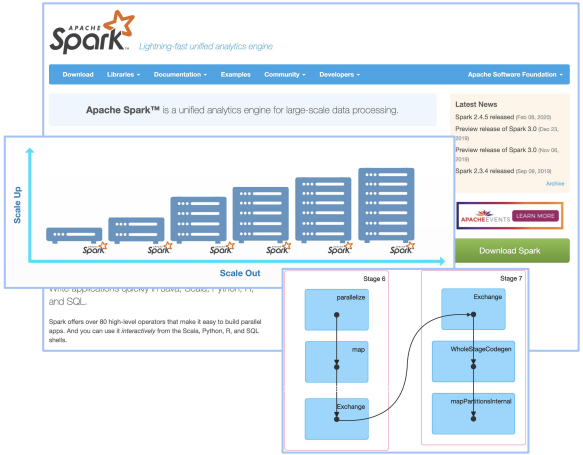
Deep Learning is a key discipline of modern AI, but Reinforcement Learning, statistical analysis and distributed computing are as relevant today.

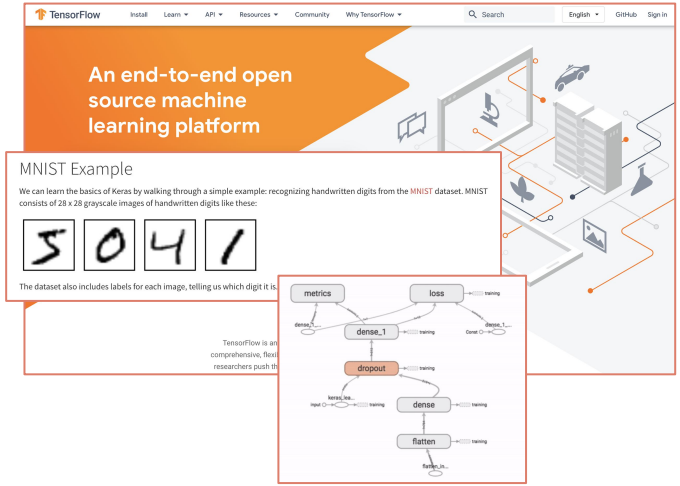


Credit: therinspark.com

Frameworks

Apache Spark, TensorFlow, PyTorch, MLflow, Horovod, and so on, are arguably required frameworks to build AI



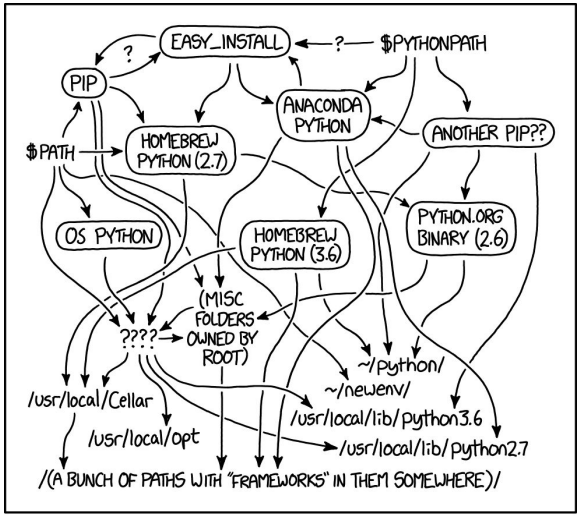


PyTorch

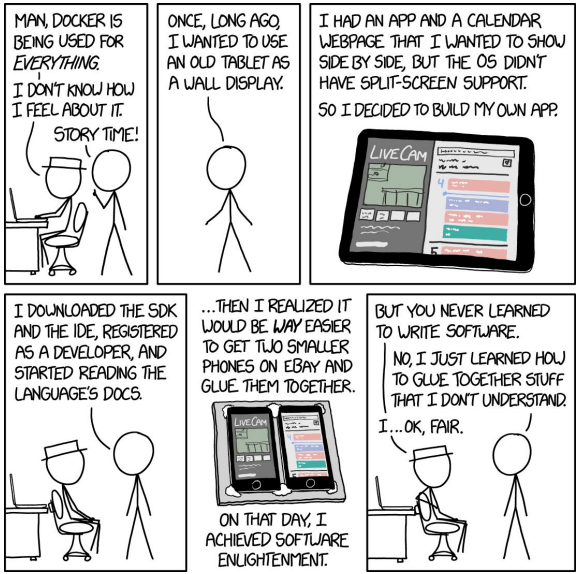
mlflow

Today

AI is still hard, requires highly-qualified teams to set up the infrastructure and successfully apply AI techniques.



MY PYTHON ENVIRONMENT HAS BECOME SO DEGRADED THAT MY LAPTOP HAS BEEN DECLARED A SUPERFUND SITE.

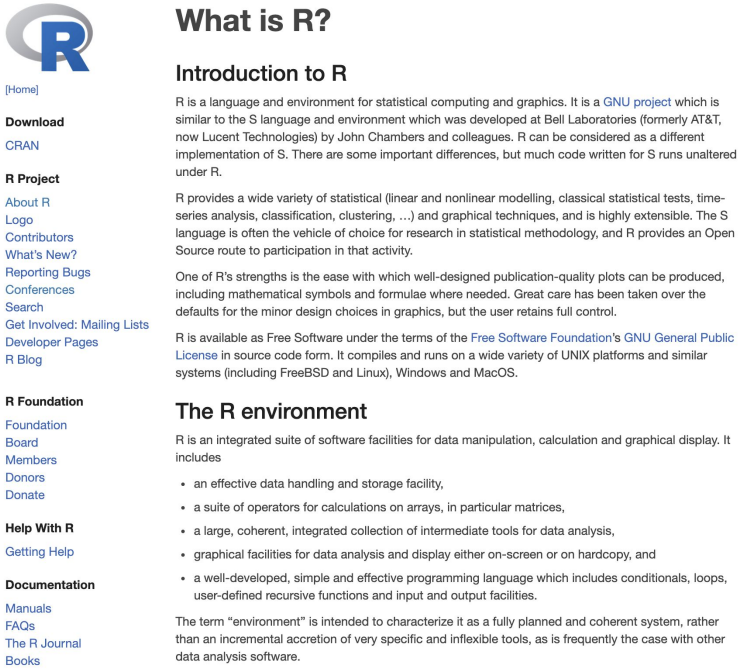


Installing and using say: TensorFlow, Spark and MLflow, is still hard.

Credit: xkcd.com

R

R is a language and environment for statistical computing and graphics, with emphasis on making Data Science accessible to everyone and growing!



[Home]

Download

CRAN

R Project

About R
Logo
Contributors
What's New?
Reporting Bugs
Conferences
Search
Get Involved: Mailing Lists
Developer Pages
R Blog

R Foundation

Foundation
Board
Members
Donors
Donate

Help With R

Getting Help

Documentation

Manuals
FAQs
The R Journal
Books

What is R?

Introduction to R

R is a language and environment for statistical computing and graphics. It is a GNU project which is similar to the S language and environment which was developed at Bell Laboratories (formerly AT&T, now Lucent Technologies) by John Chambers and colleagues. R can be considered as a different implementation of S. There are some important differences, but much code written for S runs unaltered under R.

R provides a wide variety of statistical (linear and nonlinear modelling, classical statistical tests, time-series analysis, classification, clustering, ...) and graphical techniques, and is highly extensible. The S language is often the vehicle of choice for research in statistical methodology, and R provides an Open Source route to participation in that activity.

One of R's strengths is the ease with which well-designed publication-quality plots can be produced, including mathematical symbols and formulae where needed. Great care has been taken over the defaults for the minor design choices in graphics, but the user retains full control.

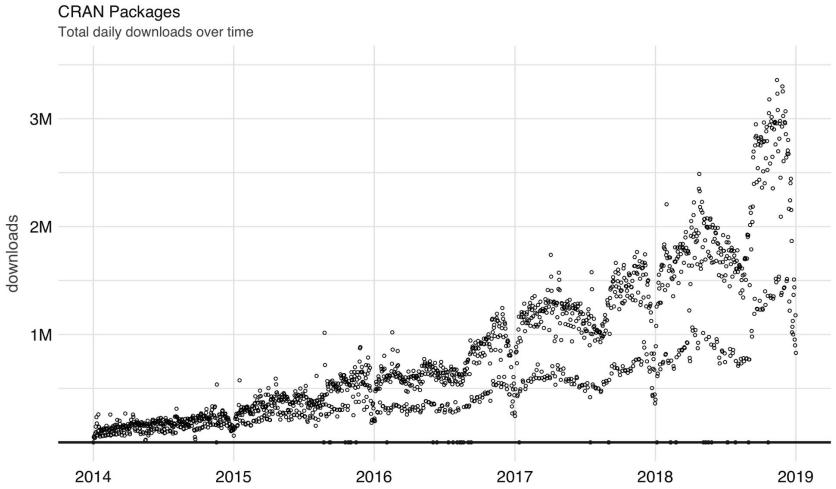
R is available as Free Software under the terms of the Free Software Foundation's GNU General Public License in source code form. It compiles and runs on a wide variety of UNIX platforms and similar systems (including FreeBSD and Linux), Windows and MacOS.

The R environment

R is an integrated suite of software facilities for data manipulation, calculation and graphical display. It includes

- an effective data handling and storage facility,
- a suite of operators for calculations on arrays, in particular matrices,
- a large, coherent, integrated collection of intermediate tools for data analysis,
- graphical facilities for data analysis and display either on-screen or on hardcopy, and
- a well-developed, simple and effective programming language which includes conditionals, loops, user-defined recursive functions and input and output facilities.

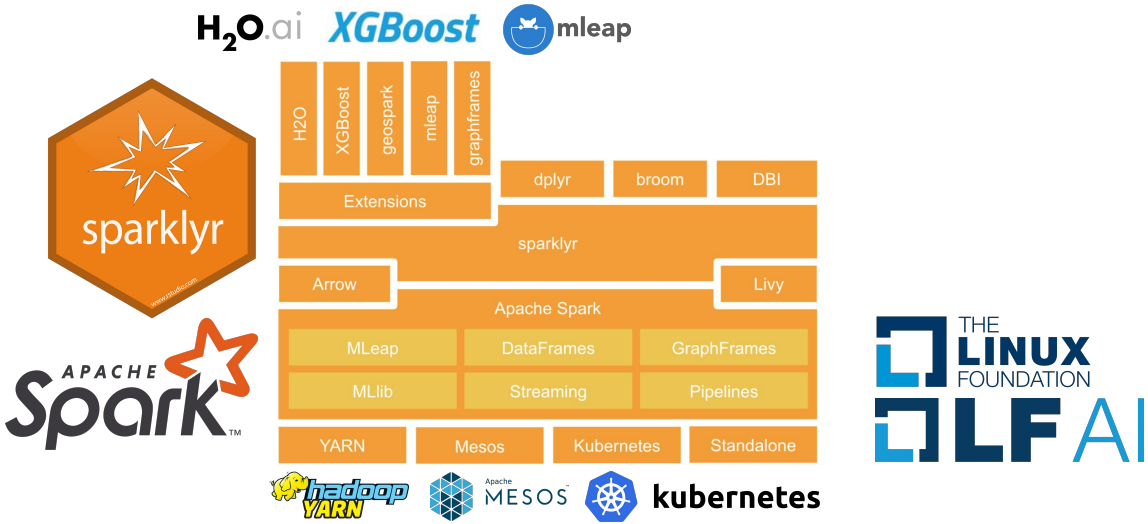
The term "environment" is intended to characterize it as a fully planned and coherent system, rather than an incremental accretion of very specific and inflexible tools, as is frequently the case with other data analysis software.



Credit: r-project.org, therinspark.com

Sparklyr

Sparklyr is an open-source and modern interface to scale data science and machine learning workflows using Apache Spark™ and R.



Sparklyr joins the Linux Foundation under LF AI in 2020.

Credit: sparklyr.ai, lfai.foundation

Ease of use

Sparklyr focuses on ease of use. You can install sparklyr and dependencies like Spark with one line of code, even in Windows:

```
# Install Spark
install.packages("sparklyr")
sparklyr::spark_install()
```

You can also then connect to local or remote Spark clusters with a line of code:

```
# Connect to Spark
sc ← sparklyr::spark_connect(master = "local")
```

Credit: sparklyr.ai

Comprehensive

```
spark_install()                                  # Install local Spark
sc <- spark_connect(master = "local")            # Connect to Spark cluster

cars <- spark_read_csv(sc, "cars", "input/")     # Read data in Spark

summarize(cars, n = n())                         # Count records with dplyr
dbGetQuery(sc, "SELECT count(*) FROM cars")      # Count records with DBI

ml_linear_regression(cars, mpg ~ wt + cyl)       # Perform linear regression

ml_pipeline(sc) %>%                              # Define Spark pipeline
  ft_r_formula(mpg ~ wt + cyl) %>%               # Add formula transformation
  ml_linear_regression()                         # Add model to pipeline

spark_context(sc) %>% invoke("version")          # Extend sparklyr with Scala
spark_apply(cars, nrow)                          # Extend sparklyr with R

stream_read_csv(sc, "input/") %>%                # Define Spark stream
  filter(mpg > 30) %>%                           # Add dplyr transformation
  stream_write_json("output/")                   # Start processing stream
```

Credit: sparklyr.ai

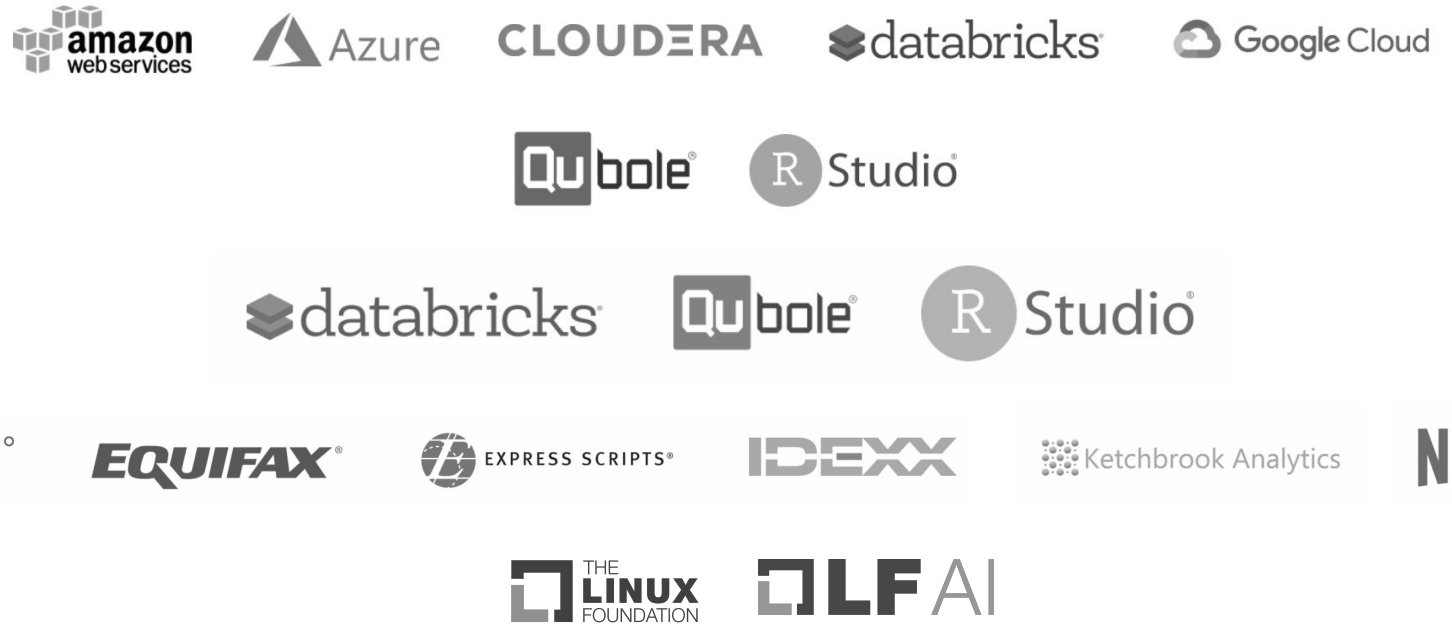
Rich in Functionality

But you can also use sparklyr to train complex models using frameworks like TensorFlow and Spark.


```r
library(sparklyr)
sc <- spark_connect(master = "local|yarn|etc")

# partition dataset
sdf_len(sc, 3, repartition = 3) %>%
  spark_apply(function(df, barrier) {
      library(tensorflow)
      library(keras)

      # define configuration from barrier
      Sys.setenv(TF_CONFIG = "")

      # define strategy and model
      strategy <- MultiWorkerMirroredStrategy()
      with (strategy$scope(), {
        model <- keras_model_sequential() # %>% ...
        model %>% compile()
      })

      # fit and retrieve model
      model %>% fit()
  }, barrier = TRUE) %>% collect()
```

Sponsors and Users

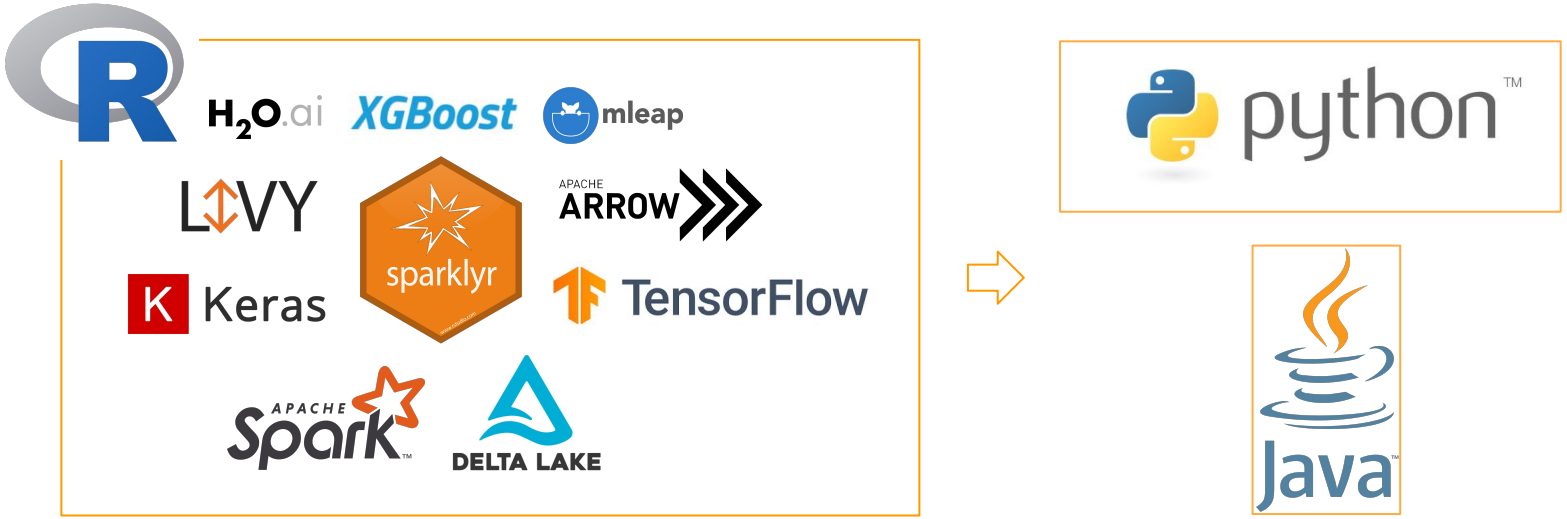
Sparklyr is support by all major cloud providers, is sponsored by Databricks, Qubole and RStudio, serves many users and is hosted within LF AI.



Credit: sparklyr.ai

Ecosystem

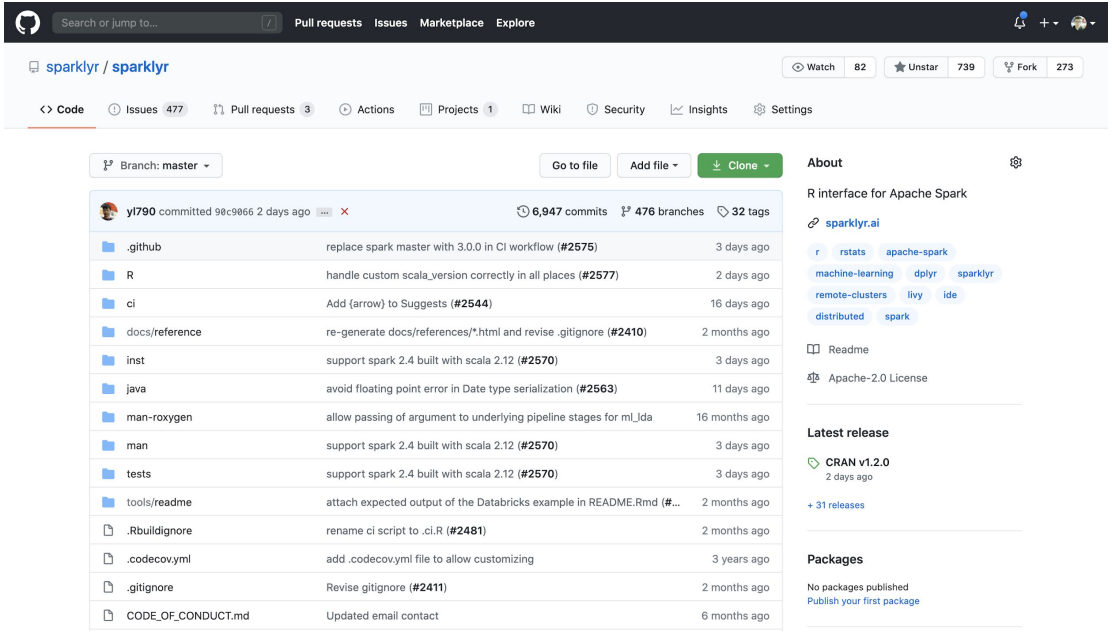
Sparklyr makes use of Apache Spark, MLlib, MLeap, Apache Livy and Apache Arrow from R, then pipelines can be used to export to Python or Java.



Local development is encouraged before running analysis in Spark clusters.

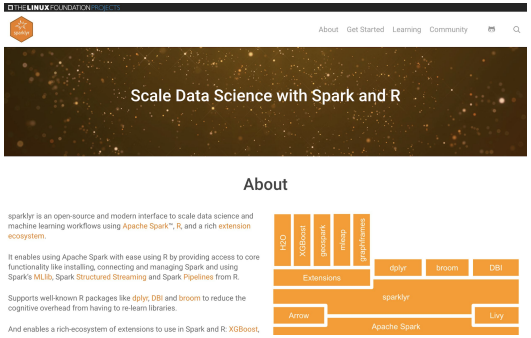
Contributing

We have dozens of major features requested and hundreds of issues, connect with us at github.com/sparklyr!

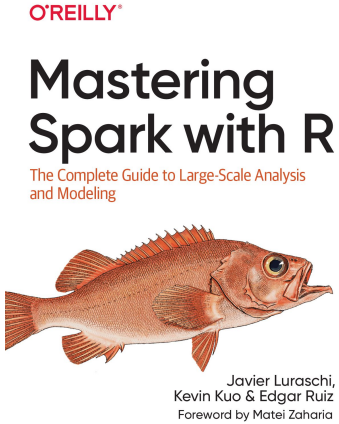


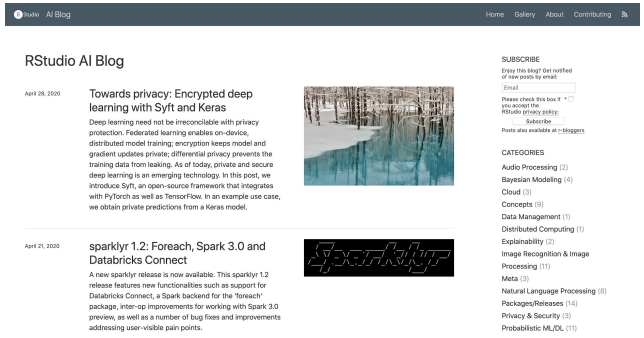
Resources

Thank you! Please learn more at sparklyr.ai, Mastering Spark with R, or thr RStudio AI blog.



sparklyr.ai





blogs.rstudio.com/ai