

Coding Machine Learning Algorithms

Jesús Fernández-Villaverde¹ and Galo Nuño²

August 21, 2023

¹University of Pennsylvania

²Banco de España

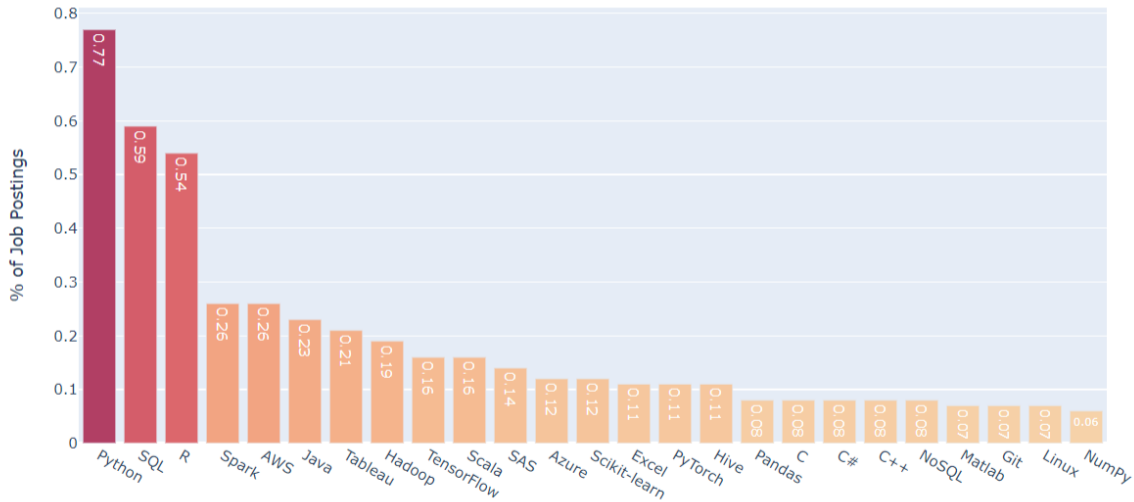
Which programming language?

- Which programming language should you use for your machine learning project?
- At a fundamental level, it does not matter. All modern scientific programming languages can implement machine learning efficiently.
- Much more important: having access to a good machine learning library (or an API to a good machine learning library). Why?
- See, for many more details, my notes on scientific programming languages.

Python: An Introduction

- Most popular language for researchers in machine learning is Python together with libraries such as PyTorch.
- Quasi-standard in the data science/machine learning industry.
- In fact, Python is the most popular programming language overall as measured by many indices.
- Influential in new generation of programming languages (Julia, Swift, ...).

25 Most In-Demand Data Science Skills in 2021








TIOBE Index for August 2022

August Headline: Python going through the roof

Python seems to be unstoppable. The scripting language gained another 2% this month. It is now at an all time high of 15.42% market share. It is hard to find a field of programming in which Python is not used extensively nowadays. The only exception is (safety-critical) embedded systems because of Python being dynamically typed and too slow. That is why the performant languages C and C++ are gaining popularity as well at the moment. If we look at the rest of the TIOBE index, not that much happened last month. Swift and PHP swapped places again at position 10, Rust is getting close to the top 20, Kotlin is back in the top 30, and the new Google language Carbon enters the TIOBE index at position 192. -- Paul Jansen CEO TIOBE Software

The TIOBE Programming Community index is an indicator of the popularity of programming languages. The index is updated once a month. The ratings are based on the number of skilled engineers world-wide, courses and third party vendors. Popular search engines such as Google, Bing, Yahoo!, Wikipedia, Amazon, YouTube and Baidu are used to calculate the ratings. It is important to note that the TIOBE index is not about the *best* programming language or the language in which *most lines of code* have been written.

The index can be used to check whether your programming skills are still up to date or to make a strategic decision about what programming language should be adopted when starting to build a new software system. The definition of the TIOBE index can be found [here](#).

Aug 2022	Aug 2021	Change	Programming Language	Ratings	Change
1	2	▲	 Python	15.42%	+3.56%
2	1	▼	 C	14.59%	+2.03%
3	3		 Java	12.40%	+1.98%
4	4		 C++	10.17%	+2.81%
5	5		 C#	5.59%	+0.45%
6	6		 Visual Basic	4.99%	+0.33%
7	7		 JavaScript	2.33%	-0.61%
8	9	▲	 Assembly language	2.17%	+0.14%
9	10	▲	 SQL	1.70%	+0.23%
10	8	▼	 PHP	1.39%	-0.80%

- Designed by Guido von Rossum in the late 1980s to emphasize readability.
 - Check *The Zen of Python*, <https://peps.python.org/pep-0020/>.
 - Easily extensible by modules.
- Name inspired by *Monty Python's Flying Circus*.
- Start working on it is straightforward with either Colab or Jupyter.
- Also, if you have a Mac, it is already preinstalled (although, likely, an older version).
- Ideal, for instance, to teach high school students or a class in introductory CS.



Table of contents

- Getting started
- Data science
- Machine learning
- More Resources
 - Featured examples
- Section

Welcome to Colab!

If you're already familiar with Colab, check out this video to learn about interactive tables, the executed code history view, and the command palette.



What is Colab?

Colab, or "Colaboratory", allows you to write and execute Python in your browser, with

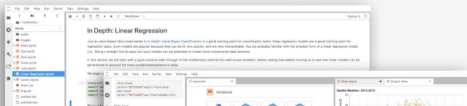
- Zero configuration required
- Access to GPUs free of charge
- Easy sharing

Whether you're a **student**, a **data scientist** or an **AI researcher**, Colab can make your work easier. Watch [Introduction to Colab](#) to learn more, or just get started below!

Getting started



Free software, open standards, and web services for interactive computing across all programming languages



JupyterLab: A Next-Generation Notebook Interface

JupyterLab is the latest web-based interactive development environment for notebooks, code, and data. Its flexible interface allows users to configure and arrange workflows in data science, scientific computing, computational journalism, and machine learning. A

- Open source.
- Current flavor: 3.10.4. A lot of old code in Python 2.x.x. Do not use it!
- High-level, general-purpose, multi-paradigm, strongly dynamically typed (with duck typing), interpreted language.
- Elegant and intuitive.
- Simple and, yet, with full OOP and functional programming support (even some metaprogramming).
- Automatic memory management.



Donate



Search

GO

Socialize

About

Downloads

Documentation

Community

Success Stories

News

Events

```
# Python 3: Fibonacci series up to n
>>> def fib(n):
>>>     a, b = 0, 1
>>>     while a < n:
>>>         print(a, end=' ')
>>>         a, b = b, a+b
>>>     print()
>>> fib(1000)
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987
```



Functions Defined

The core of extensible programming is defining functions. Python allows mandatory and optional arguments, keyword arguments, and even arbitrary argument lists. [More about defining functions in Python 3](#)

- 1
- 2
- 3
- 4
- 5

Python is a programming language that lets you work quickly and integrate systems more effectively. [>>> Learn More](#)

Contribute today to the Python PSF Spring Fundraiser and show your love for Python!

Donate Now

Get Started

Whether you're new to programming or an experienced developer, it's easy to learn and use

Download

Python source code and installers are available for download for all versions!

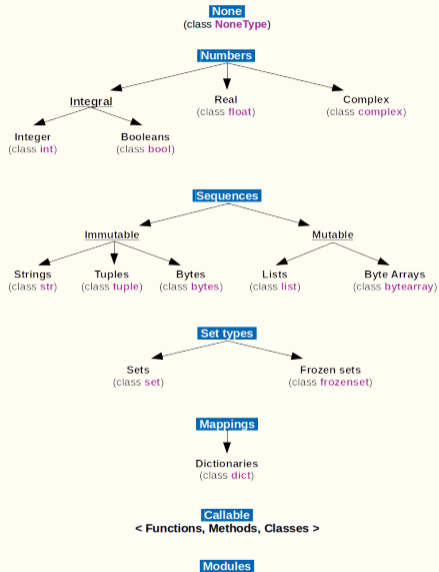
Docs

Documentation for Python's standard library, along with tutorials and guides, are available online

Jobs

Looking for work or have a Python related position that you're trying to hire for? Our **relaunched**

Python 3 The standard type hierarchy



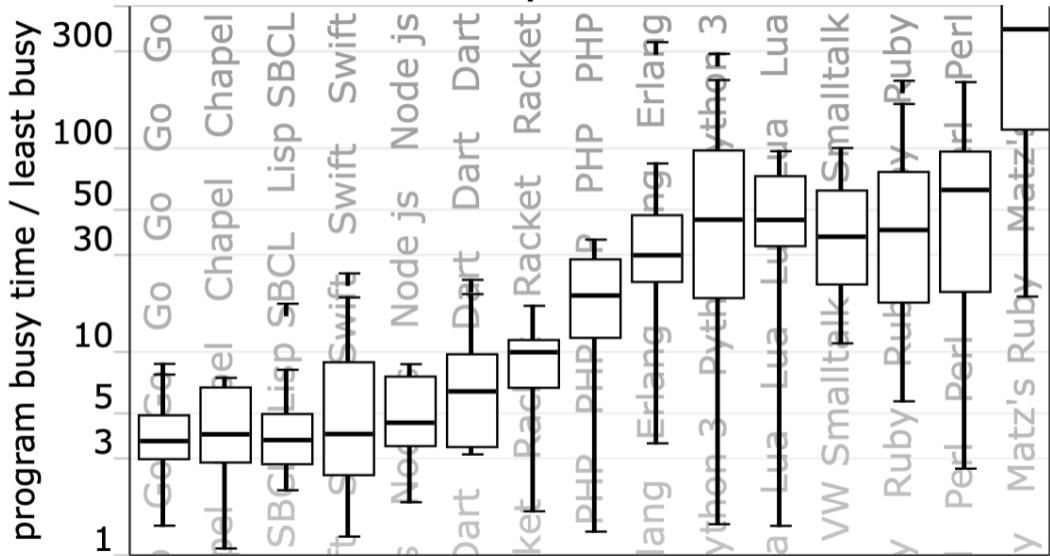
Python: advantages

1. Great for prototyping: dynamic typing and REPL.
2. Rich ecosystem:
 - 2.1 Scientific computation modules: NumPy, SciPy, and SymPy.
 - 2.2 Statistics modules: Pandas.
 - 2.3 Plotting modules: matplotlib and ggplot.
3. Easy unit testing: doctest is a default module.
4. Manipulates strings surprisingly well (regular expressions) ⇒ natural language processing, artificial intelligence, big data.
5. Excellent interaction with other languages.

Python: disadvantages

1. Since it is an interpreted language, Python pays a considerable time penalty.
2. Most machine learning libraries, though, are coded in the backend in C++.
3. Also, Numba is a just-in-time specializing compiler which compiles annotated Python and NumPy code to LLVM.
4. Other compilation possibilities are more cumbersome.
5. All assignments are by object, which can look confusing to those without experience.
6. (Personal preference): matrix indexing starts at zero.

How many times slower?





Numba

Numba makes Python code fast

Numba is an open source JIT compiler that translates a subset of Python and NumPy code into fast machine code.

[Learn More](#)

[Try Numba »](#)

Python: Libraries for machine learning

Python: Libraries for machine learning

- As mentioned before, libraries are key.
- Four main options:
 1. Scikit: easy for basic ML algorithms, in particular for data analysis.
 2. Tensorflow: Being phased out.
 3. PyTorch: most popular in research, object-oriented, faster. Also, Pyro is a probabilistic DSL built on Python and PyTorch.
 4. Jax: innovative, functional programming.
 5. Keras: Popular multi-backend.





TensorFlow

FROM RESEARCH TO PRODUCTION

An open source machine learning framework that accelerates the path from research prototyping to production deployment.

Install >

< NEW! PyTorch 1.11, TorchData, and functorch are now available >

KEY FEATURES & CAPABILITIES

See all Features >

Support Ukraine 🇺🇦
Help Provide Humanitarian Aid to Ukraine.

Production Ready

Transition seamlessly

Distributed Training

Scalable distributed

Robust Ecosystem

A rich ecosystem of tools

Cloud Support

PyTorch is well supported



GETTING STARTED

[Installing JAX](#)[JAX Quickstart](#)[How to Think in JAX](#)[JAX - The Sharp Bits](#)[Tutorial: JAX 101](#)

REFERENCE DOCUMENTATION

[JAX Frequently Asked Questions \(FAQ\)](#)[Asynchronous dispatch](#)[Understanding Jaxprs](#)[Convolutions in JAX](#)[Pytrees](#)[Type promotion semantics](#)[JAX Errors](#)[JAX Glossary of Terms](#)[Change log](#)

ADVANCED JAX TUTORIALS

[The Autodiff Cookbook](#)[Contents](#)[JAX reference documentation](#)[Indices and tables](#)

JAX reference documentation

JAX is [Autograd](#) and [XLA](#), brought together for high-performance numerical computing and machine learning research. It provides composable transformations of Python+NumPy programs: differentiate, vectorize, parallelize, Just-In-Time compile to GPU/TPU, and more.

Getting Started

- [Installing JAX](#)
- [JAX Quickstart](#)
- [How to Think in JAX](#)
- [JAX - The Sharp Bits](#)
- [Tutorial: JAX 101](#)
 - [JAX As Accelerated NumPy](#)
 - [Just In Time Compilation with JAX](#)
 - [Automatic Vectorization in JAX](#)
 - [Advanced Automatic Differentiation in JAX](#)
 - [Pseudo Random Numbers in JAX](#)
 - [Working with Pytrees](#)
 - [Parallel Evaluation in JAX](#)
 - [Stateful Computations in JAX](#)
 - [Introduction to pjit](#)

Reference Documentation

- [JAX Frequently Asked Questions \(FAQ\)](#)
- [Asynchronous dispatch](#)
- [Understanding Jaxprs](#)
- [Convolutions in JAX](#)



Keras

O'REILLY®

Hands-on Machine Learning with Scikit-Learn, Keras & TensorFlow

Concepts, Tools, and Techniques
to Build Intelligent Systems

powered by



Aurélien Géron

2nd Edition
Updated for
TensorFlow 2



Aurélien Geron
ageron

Follow

Author of the book Hands-On Machine Learning with Scikit-Learn and TensorFlow. Former PM of YouTube video classification and founder & CTO of a telco operator.

10.6k followers · 4 following

Popular repositories

hands-on-ml Public

A series of Jupyter notebooks that walk you through the fundamentals of Machine Learning and Deep Learning in python using Scikit-Learn and TensorFlow.

Jupyter Notebook ☆ 24.4k 🍷 12.8k

hands-on-ml2 Public

A series of Jupyter notebooks that walk you through the fundamentals of Machine Learning and Deep Learning in Python using Scikit-Learn, Keras and TensorFlow 2.

Jupyter Notebook ☆ 20.8k 🍷 10.3k

tf2_course Public

Notebooks for my "Deep Learning with TensorFlow 2 and Keras" course

Jupyter Notebook ☆ 1.8k 🍷 504

hands-on-ml3 Public

A series of Jupyter notebooks that walk you through the fundamentals of Machine Learning and Deep Learning in Python using Scikit-Learn, Keras and TensorFlow 2.

Jupyter Notebook ☆ 237 🍷 62

tensorflow-safari-course Public

DEPRECATED — Exercises and solutions to accompany my Safari course introducing TensorFlow.

Jupyter Notebook ☆ 170 🍷 115

tiny-dqn Public

A tiny implementation of Deep Q Learning, using TensorFlow and OpenAI gym

Python ☆ 92 🍷 45

DEEP LEARNING with Python

SECOND EDITION

François Chollet

 MANNING



Deep Learning with PyTorch

Eli Stevens
Luca Antiga
Thomas Viehmann
Foreword by Sumit Chintala

 MANNING



Python: Setting up a workflow

Python: Setting up a workflow

- You can work online.
- But, eventually, you might want to code on your computer (even if you run the code on the cloud).
- Get yourself either a good editor or a good IDE.
- Get yourself a good basic textbook and some extra books for HPC and machine learning.
- Some basic tutorials at: <https://software-carpentry.org/lessons/>.

code.visualstudio.com

Click to go back, hold to see history

visual Studio Code Docs Updates Blog API Extensions FAQ Learn

Search Docs

Download

Version 1.67 is now available! Read about the new features and fixes from April.

Code editing. Redefined.

Free. Built on open source. Runs everywhere.

Download Mac Universal
Stable Build

Other platforms and Insiders Edition

By using VS Code, you agree to its
license and privacy statement.

The screenshot shows the Visual Studio Code interface. On the left is the 'EXTENSIONS: MARKETPLACE' sidebar with a list of extensions including Python, GitLens, C/C++, ESLint, Debugger for Chrome, Language Support for Java, and Vetur. The main editor area shows a code file named 'blog-post.js' with a React component structure. The terminal at the bottom shows compilation output: 'info [win32]: Compiling... [Docker] Compiled successfully in 26ms' and 'info [win32]: info [win32]: Compiled successfully.'



IntelliSense



Run and Debug



Built-in Git



Extensions



PyCharm

The Python IDE
for Professional Developers

DOWNLOAD

Full-fledged Professional or Free Community

2ND EDITION

PYTHON CRASH COURSE

A HANDS-ON, PROJECT-BASED
INTRODUCTION TO PROGRAMMING

ERIC MATTHES

OVER
1,000,000
COPIES SOLD



Powerful Object-Oriented Programming

5th Edition
Updated for 3.3 and 2.7

Learning

Python



O'REILLY*

Mark Lutz

O'REILLY®

Second
Edition

High Performance Python

Practical Performant
Programming for Humans



Micha Gorelick & Ian Ozsvald

Alternatives

Other languages

- My favorite language is Julia. Why?
- R can be a good alternative if you come from econometrics/data science. Also, RStudio is one of the best IDEs available.
- Problems with Matlab. Why?
- Please, do not consider Stata or Fortran.
- I will have examples from Matlab and Python.



[Download](#)

[Documentation](#)

[Blog](#)

[Community](#)

[Learn](#)

[Research](#)

[JSoC](#)

[♥ Sponsor](#)

The Julia Programming Language

[Download](#)

[Documentation](#)

★ Star 39,090

Julia in a Nutshell

Fast

Julia was designed from the beginning for [high performance](#). Julia programs compile to efficient native code for [multiple platforms](#) via LLVM.

Dynamic

Julia is [dynamically typed](#), feels like a scripting language, and has good support for [interactive](#) use.

Reproducible

[Reproducible environments](#) make it possible to recreate the same Julia environment every time, across platforms, with [pre-built binaries](#).

Composable

General

Open source

O'REILLY®

Think Julia

How to Think Like a Computer Scientist



Ben Lauwens &
Allen B. Downey



Flux

Search docs

Home

- Installation
- Learning Flux

Building Models

- Overview
- Basics
- Recurrence
- Model Reference
- Loss Functions
- Regularisation
- Advanced Model Building
- NNlib
- Functors

Handling Data

- One-Hot Encoding
- MLUtils

Version v0.13.0 ▾

Home

[Edit on GitHub](#) ⚙️

Flux: The Julia Machine Learning Library

Flux is a library for machine learning geared towards high-performance production pipelines. It comes "batteries-included" with many useful tools built in, but also lets you use the full power of the Julia language where you need it. We follow a few key principles:

- Doing the obvious thing. Flux has relatively few explicit APIs for features like regularisation or embeddings. Instead, writing down the mathematical form will work – and be fast.
- Extensible by default. Flux is written to be highly extensible and flexible while being performant. Extending Flux is as simple as using your own code as part of the model you want - it is all [high level Julia code](#). When in doubt, it's well worth looking at [the source](#). If you need something different, you can easily roll your own.
- Performance is key. Flux integrates with high-performance AD tools such as [Zygote.jl](#) for generating fast code. Flux optimizes both CPU and GPU performance. Scaling workloads easily to multiple GPUs can be done with the help of Julia's [GPU tooling](#) and projects like [DaggerFlux.jl](#).
- Play nicely with others. Flux works well with Julia libraries from [data frames](#) and [images](#) to [differential equation solvers](#), so you can easily build complex data processing pipelines that integrate Flux models.

Installation

Download [Julia 1.0](#) or later, if you haven't already. You can add Flux from using Julia's package manager, by typing `] add Flux` in the Julia prompt.

If you have CUDA you can also run `] add CUDA` to get GPU support; see [here](#) for more details.

NOTE: Flux used to have a CuArrays.jl dependency until v0.10.4, replaced by CUDA.jl in v0.11.0. If you're upgrading Flux from v0.10.4 or a lower version, you may need to remove CuArrays (run `] rm CuArrays`) before you can upgrade.

Learning Flux

There are several different ways to learn Flux. If you just want to get started writing models, the [model](#)

SECOND EDITION



R

IN ACTION

Data analysis and graphics with R

Robert I. Kabacoff

 MANNING

The R Series

Hands-On Machine Learning with R



Bradley Boehmke
Brandon Greenwell

 CRC Press
Taylor & Francis Group
A CHAPMAN & HALL BOOK