Seminario PYTHON - MCAF

Francisco Gárate Santiago

UCM, 2021

Sobre mi
Francisco Gárate

- ▶ Licenciado en ADE
- ▶ Licenciado en Ciencias Actuariales y Financieras (CAF)
- ▶ Actuario colegiado
- ▶ Ponente de la EPAF (Escuela de Práctica Actuarial y Financiera) del Instituto de Actuarios Españoles
- ▶ Autor de la librería de python pyliferisk (cálculo actuarial)
- ▶ Responsable de la función actuarial del grupo ASISA
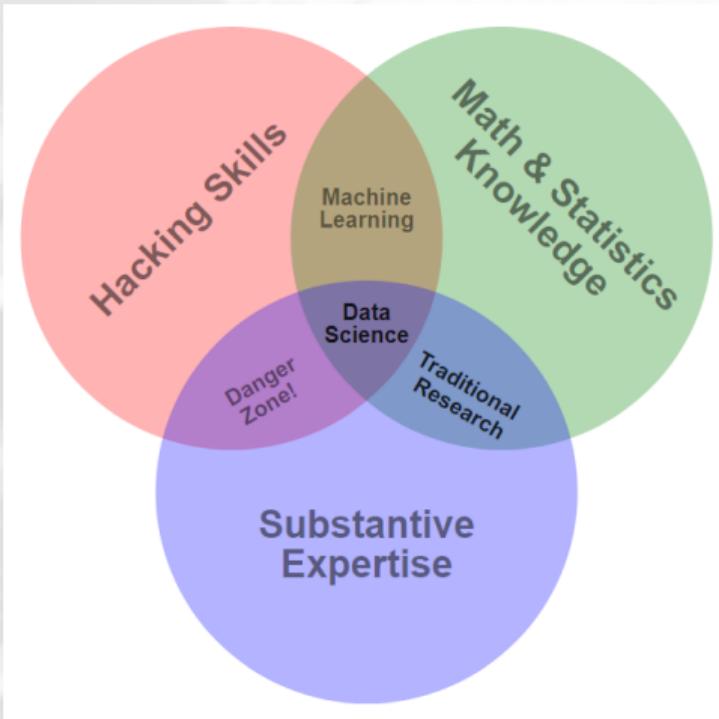
Objetivo

Programación en lenguaje Python

- ► Conocer la importancia de adquirir conocimientos en el principal lenguaje de programación en el campo de la ciencia de datos, big data y aprendizaje automático.
- ► Familiarizarse con el lenguaje de programación python. Ser capaz de crear o modificar una función en python, realizar bucles y condicionales.
- ► Conocer las principales funciones y utilidades de las principales librerías estadísticas: numpy, scipy y pandas.
- ► Introducción al aprendizaje automático con el lenguaje python.

¿Qué es la Ciencia de Datos?

Data Science Venn Diagram - Drew Conway, 2010

- ► **Habilidades Hacker**. Ser capaz de manipular archivos de texto en la línea de comandos, comprender operaciones vectorizadas, pensamiento abstracto, identificación de patrones ganadores...
- ► **Conocimientos matemáticos y estadísticos**: familiaridad básica con estas herramientas, ejemplo: qué es una regresión de mínimos cuadráticos ordinarios y cómo interpretarla.
- ► **Experiencia real**, tener toda la información y ademas saber que hacer con ella, saber enfrentarse a casos reales. (No vale el: *Supongamos una vaca esférica*).

¿Por qué Python?

▶ Los lenguajes interpretados (como R, Python o Julia) consisten en líneas de instrucciones o scripts que son interpretados en tiempo real, por tanto no requieren ser compilados (como si sucede con C o C++) y pueden ser ejecutados en cualquier sistema operativo e incluso bajo entornos web.

▶ Python es uno de los lenguajes más utilizados en la denominada ciencia de datos (*data science*). Es fácil de escribir y de ser entendido sin necesidad de tener conocimientos avanzados de programación.

▶ Posee una de las mayores comunidades de desarrolladores (principalmente *open-source*), siendo uno de los lenguajes más populares en las comunidades de soporte (Stackoverflow, Github...), muy por encima de cualquier otro lenguaje.

▶ Python tiene soporte nativo en las principales bases de datos (SQL, DB2, Oracle, SAP DB...) y así como aquellas bases de datos utilizadas en Big Data (Azure SQL, Hadoop, Apache Beam, MongoDB o CDH).

¿Por qué Python?

`https://bigpanda.cern.ch`, `http://opendata.cern.ch`

▶ Desarrolladores: CERN, NASA, Federal Reserve Bank, Facebook...

Python @ CERN

Sebastian Witowski

CERN – home of the Large Hadron Collider that can spit up to **1 petabyte of collision data per second**, impossible for the computing systems to record. The experiments, however, need only a small fraction of "interesting" events. The first-level trigger of ATLAS, for example, selects only **1 in 400 events** (making decisions in ~2.5 millionths of a second). These are sent at **160 GB/second** to the tens of thousands of processor cores which select **1%** of the remaining events for analysis. Even after such drastic data reduction, there are still around **50 petabytes** of data produced at CERN per year.

How does Python fit in this ecosystem? It might not be fast enough to be used for filtering this amount of data, but nevertheless, there are many great projects created with Python at CERN.

python + ROOT

Thousands of scientists every day analyse data produced by the experiments at CERN. They need a tool that is based on a flexible and expressive language (like Python), but also works fast with large amount of data. And this is exactly what PyROOT is – a Python module that allows users to interact with ROOT (a data analysis framework written in C++, that is very popular in High Energy Physics community).

PyROOT is more than just a wrapper around ROOT. Python bindings are based on the C++ reflection information stored in the ROOT type system. Thanks to that, Python classes are created dynamically when needed and C++ functions and globals are available automatically in Python.

Users can use PyROOT directly in Python REPL. With all the ROOT classes and functions available out of the box, they can combine them with other modules like NumPy and SciPy to make make data analysis easier and faster.
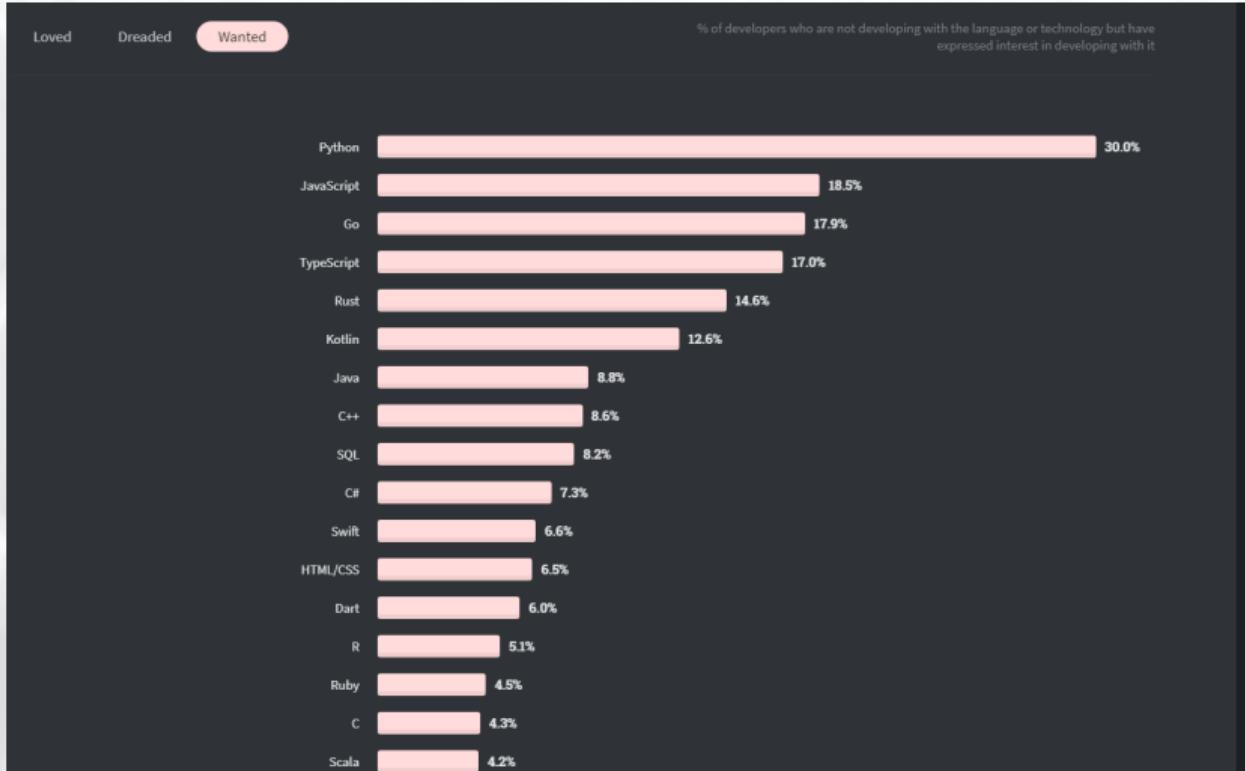
SWAN (Service for Web based Analysis) allows scientists to analyse their data in the CERN cloud. It uses Jupyter notebooks as the interface. The users' sessions are encapsulated in containers, synchronized and stored via CERNBox (file synchronization and sharing service build on top of Owncloud). Large datasets can be accessed on EOS - the disk-based storage service of CERN.

¿Por qué Python?

https://insights.stackoverflow.com/survey/2020

	Loved	Dreaded	Wanted	% of developers who are not developing with the language or technology but have expressed interest in developing with it

Python	30.0%
JavaScript	18.5%
Go	17.9%
TypeScript	17.0%
Rust	14.6%
Kotlin	12.6%
Java	8.8%
C++	8.6%
SQL	8.2%
C#	7.3%
Swift	6.6%
HTML/CSS	6.5%
Dart	6.0%
R	5.1%
Ruby	4.5%
C	4.3%
Scala	4.2%
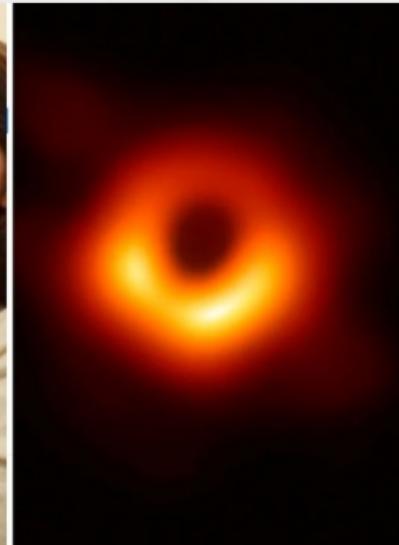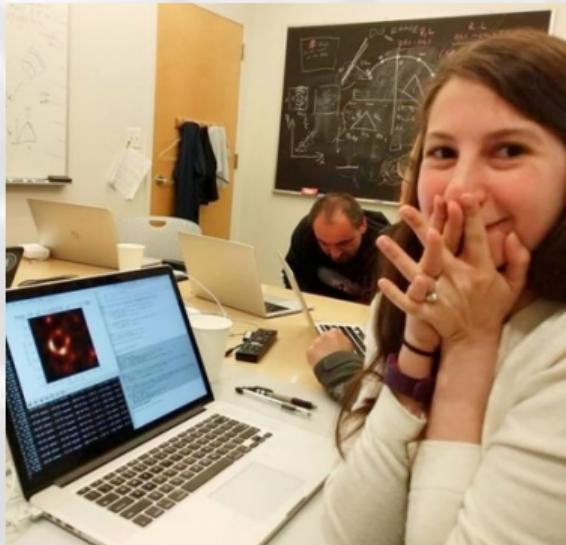
¿Por qué es importante saber programar?

Iras más lejos

Katie Bouman, (MIT)
`https://numpy.org/case-studies/blackhole-image/`
`https://analyticsindiamag.com/`
`how-imaging-a-blackhole-gives-us-one-more-reason-to-embrace-python-for-larger-datasets/`

¿Por qué es importante saber programar?

Con menor esfuerzo

Nihad Abbasov,

`https://github.com/NARKOZ/hacker-scripts`

¿Qué es la Ciencia de Datos?

Unicornios

https://www.tesla.com/careers/search/job/revolutionary-actuary-71528
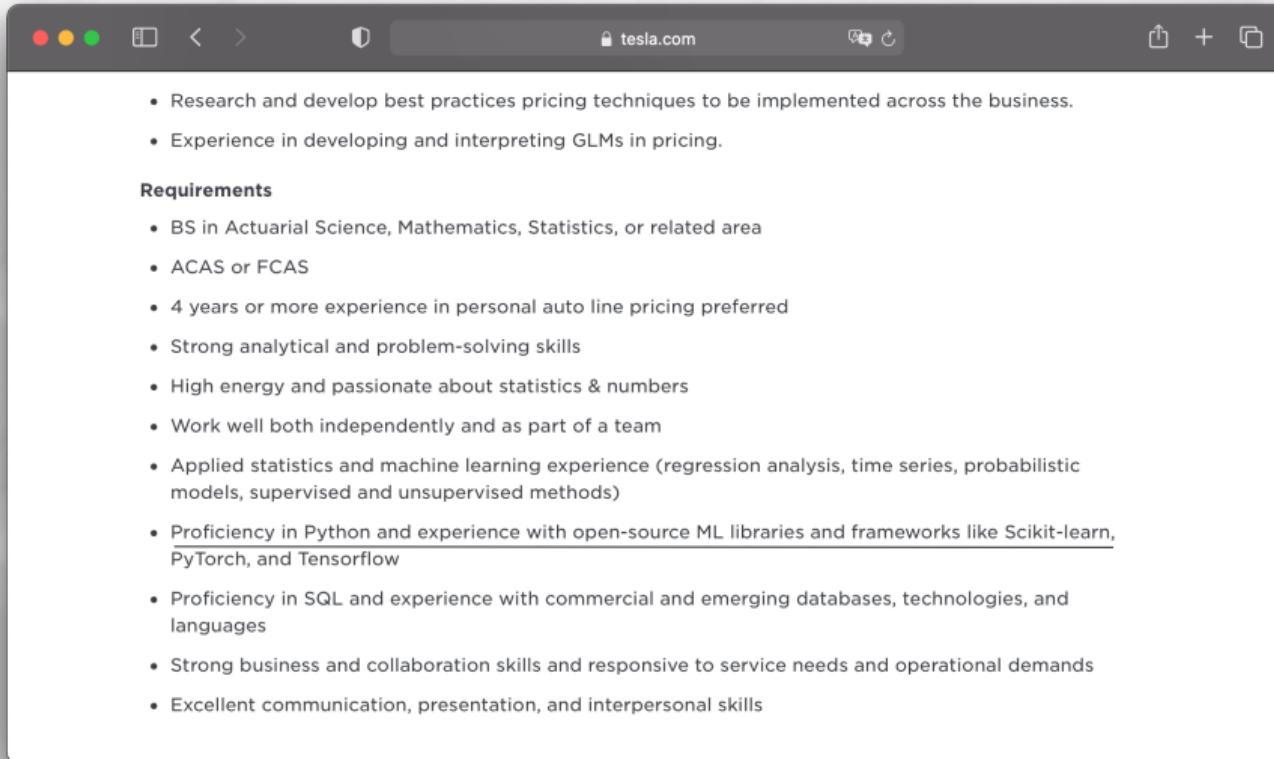
¿Qué es la Ciencia de Datos?
Unicornios

- Research and develop best practices pricing techniques to be implemented across the business.
- Experience in developing and interpreting GLMs in pricing.

Requirements

- BS in Actuarial Science, Mathematics, Statistics, or related area
- ACAS or FCAS
- 4 years or more experience in personal auto line pricing preferred
- Strong analytical and problem-solving skills
- High energy and passionate about statistics & numbers
- Work well both independently and as part of a team
- Applied statistics and machine learning experience (regression analysis, time series, probabilistic models, supervised and unsupervised methods)
- Proficiency in Python and experience with open-source ML libraries and frameworks like Scikit-learn, PyTorch, and Tensorflow
- Proficiency in SQL and experience with commercial and emerging databases, technologies, and languages
- Strong business and collaboration skills and responsive to service needs and operational demands
- Excellent communication, presentation, and interpersonal skills

¿Qué es la Ciencia de Datos?

Unicornios

Introducción al lenguaje Python

Caracteristicas

- Los lenguajes interpretados (como R, Python o Julia) consisten en líneas de instrucciones o scripts que son interpretados en tiempo real, por tanto no requieren ser compilados (como si sucede con C o C++) y pueden ser ejecutados en cualquier sistema operativo e incluso bajo entornos web.
- Suele decirse que Python es la navaja suiza de los lenguajes de programación.
- La instalación de paquetes no puede ser más fácil. Ejemplo: ejecutar en la consola `pip install pandas` para instalar la librería/biblioteca `pandas`.

Introducción al lenguaje Python

Guido van Rossum, 1990

El lenguaje Python fue creado por el holandés Guido van Rossum en 1990. La mentalidad con qué fue creado hace que sea muy intuitivo y uno de los lenguajes más fáciles de aprender.

Introducción al lenguaje Python

Historia

Guido van Rossum ✔
@gvanrossum

Today's Python history lesson: Python took its control and data structures from ABC, its identifiers, strings and %-string formats from C, and its regular expressions from Perl. But its # comments (and #!) and -c command line flag came from the UNIX v7 shell.

Traducido del inglés al Google

Lección de historia de Python de hoy: Python tomó sus estructuras de control y datos de ABC, sus identificadores, cadenas y formatos de% de cadena de C, y sus expresiones regulares de Perl. Pero sus # comentarios (y #!) y el indicador de línea de comando -c vino del shell de UNIX v7.

7:58 a. m. · 30 ene. 2020 · Twitter Web App