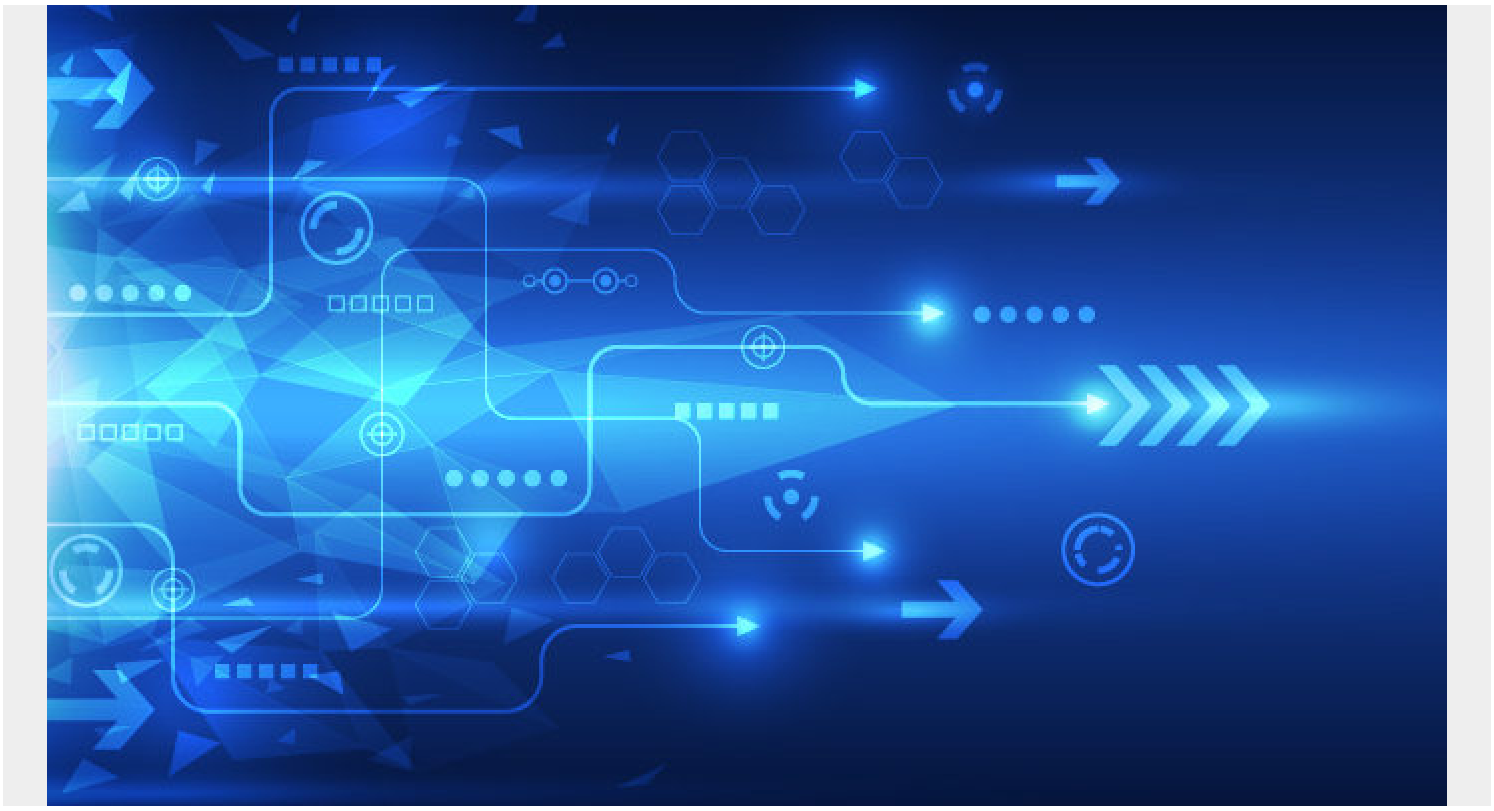
USING PYTHON FOR BIG DATA & ANALYTICS (PYTHON IS PERFECT FOR BIG DATA)



If we think of that common programming language, Python is the one that immediately comes to mind. Python has rapidly gained popularity in the IT community as a simple yet feature-rich language powering anything from simple web applications to the IoT, game development, and even artificial intelligence.

Big data and data analytics are another sector in which Python is currently making inroads. In this article, let's find out why Python is used in Big Data and Analytics.

Why use Python for big data & analytics?

When it comes to data analytics or data science, its sheer complexity becomes a major concern. So much so that you might think you need a specialized programming languages to handle such tasks. Of course, there are indeed programming languages that specialize in those fields—R, Scala, Julia, etc.

Yet, in most cases, data analytics professionals and data scientists actually prefer Python over these other languages! There are several reasons behind this popularity, let's have a look.

(Compare Python to Java & Go.)

Ease of use

Python is relatively easy to learn and much less wordy compared to other languages like Java. (This characteristic is known as being Pythonic.) This simplicity lowers the barrier to enter Python as a new programming language.

The best part is that the simplicity of the language does not affect the functionality in any shape or form, and Python is always powerful. Simply install the language, and you are ready to get started. There are no complex configurations required, such as setting up compilers.

Additionally, Python is not restricted to a particular programming style. Therefore, users are free to choose a programming methodology that they are comfortable with:

- Object oriented programming (OOP)
- Functional programming
- Etc.

Finally, the Python language consists of more readable code, which in turn helps users easily understand codebases. Python can also act as the gateway for big data and data science fields without having to learn a new language.

(New to Python? Get to work with this Python starter kit.)

Licensing structure

Python is an open-source language managed by the non-profit Python Software Foundation. This open-source nature allows Python to be used in any project without fearing any interference from a third party. Contrast this to other programming languages managed or owned by commercial organizations, where a single decision can cripple the usage of the language.

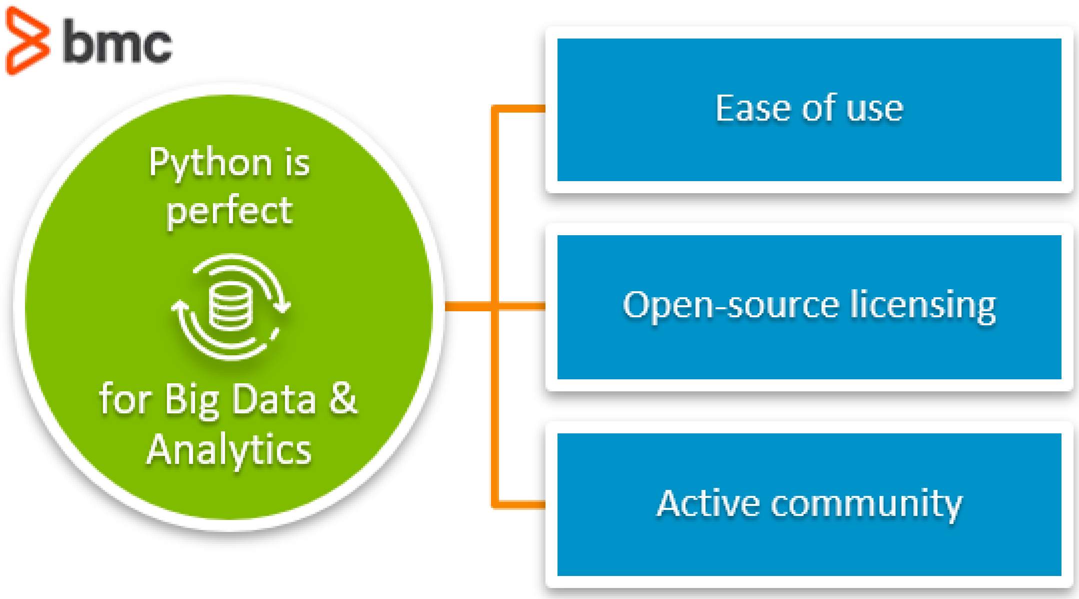
Data analytics projects can be complex and time-consuming endeavors. Thus, the open-source nature of Python helps data scientists and analysts confidently use it for the foreseeable future of any kind of commercial or hobbyist project.

Active community

Python has an active and thriving community, members of which are routinely and actively:

- Contributing to Python language development.
- Creating and maintaining thousands of packages that help enhance and extend core Python functionality.

Thanks to this large community, there are many online resources, GitHub/GitLab projects, forums, and communities available to anyone. Furthermore, there is a high chance that you will be able to find solutions to most issues by simply googling.



Python tools for Big Data and Analytics

So far, we have covered the general characteristics of Python that make it an ideal choice for Big Data and Analytics. Now let's dig a bit deeper and explore the tools available in Python to cater to these needs.

Python packages

Python comes with the Python Package Index (PIP), the open-source repository that contains all the third-party packages available for Python. This library consists of packages to help users in various tasks, from simple tasks like JSON parsing to complete data transformation, analytics, and visualizations packages.

Let's look at some of these packages.

The Pandas data analytics library is one of the most prominent packages out of these packages. Pandas have become the defector standard for both novice and experienced users in data science, engineering, and analytics fields. Furthermore, it can handle any kind of tabular data sets, enabling users to explore, clean, transform, filter, and process data.

Pandas is versatile enough to handle any data set with its ability to import and export data to and from various sources easily. Additionally, Pandas can be used to create plots, handle time-series data and textual data. All these facts have led Pandas to become one of the most powerful libraries available for data analytics.

(Explore our hands-on Pandas Guide.)

Next up: NumPy (Numerical Python) is there if you need a package to handle scientific computing and mathematical functions. NumPy provides a powerful multidimensional array object and a whole bunch of routines to manipulate arrays such as mathematical, statistical, basic linear algebra, sorting, logical functions, etc.

Due to the vectorized code, NumPy can process data faster than any other library, which is highly beneficial when dealing with complex data sets. The NumPy API is also used extensively to enhance functionality in other packages, such as

- Pandas
- SciPy
- scikit-learn
- And more

(Start with these NumPy basics.)

Visualizing data is as important as analytics. In Python, we have powerful packages such as matplotlib, seaborn, folium, and even ploty that offer comprehensive visualization capabilities. For example, the matplotlib package not only allows users to create static visualization but also offers animated interactive visualization with full control over all the aspects of the visualization. These functions can be extended further by integrating third-party packages like base map, cartopy, etc.

In addition to the packages mentioned above, there are numerous other packages geared towards data analytics and processing, such as Polaris, Desk, Vaex, PySpark, etc.

(Explore our Data Visualization Guide, which explores many of these tools & packages.)

Anaconda

Anaconda Data Science Toolkit, or simply Anaconda, is an open-source toolkit that aims to facilitate a complete data science platform with a single installation. Everything is seamlessly integrated with Anaconda, besides setting up a Python environment and installing and managing packages separately.

Anaconda provides the conda package and environment management system, built-in machine learning, data science, and visualization libraries under a single software package. In addition to Python support, Anaconda also supports R allowing users to utilize the same familiar toolkit with a specialized data science programming language.

On top of that, Anaconda Navigator users get a complete GUI to manage packages, environments, and software without relying on the command line. Anaconda supports various applications like Spyder, JupyterLab, Datalore, etc., which can also be managed via the GUI. All these features are offered for free with the Anaconda individual edition, which is distributed free of charge. There is also the option to upgrade to even more feature-rich team and enterprise editions for advanced enterprise data analytics requirements.

Python notebooks

Jupyter Notebooks, have become the defector standard when it comes to creating notebooks. Notebooks offer users a browser-based coding environment that can be used to create and share notebooks that contain everything you need in one spot:

- Code
- Visualizations
- Equations
- Text

Notebooks offer a simpler development experience and a perfect test platform without the need for setting up dedicated environments. This is highly useful when dealing with visualization as some will require complex configurations or additional addons to display the generated charts, plots, etc.

With notebooks, users can analyze, document, and visualize data in a single canvas without having to depend on multiple tools. Notebooks are not limited to Python, and they support over 40 programming languages. This extensive support allows notebooks to easily provide extended functionality using different languages in conjunction with a Python notebook.

Machine learning algorithms

All major machine learning algorithms—including TensorFlow, Microsoft Cognitive Toolkit, scikit-learn, and Spark ML—are written in Python. All these projects are continuously improved by each organization and the massive academic and hobbyist communities backing them. Therefore, users can simply utilize these algorithms and libraries in their projects to provide analytics and even build neural networks.

All the above-mentioned software, libraries, etc., are specifically built for Python or consider Python a first-class citizen, allowing users to tackle complex data analytics tasks more easily.

(Check out our ML algorithm beginner's guide.)

We love Python for big data

In this article, we had a look at why Python is used for Big Data and Analytics. Certain features of Python, such as the low barrier to get started with the language, simplicity, and licensing structure, makes it best suited for handling data science and analytics tasks.

On top of that, Python comes with a complete feature set that can be adapted to any need, coupled with all the available data analytics and data science tools proving itself one of the best languages to harness the power of big data.

Related reading

- BMC Machine Learning & Big Data Blog
- Data Architecture Explained: Components, Standards & Changing Architectures
- Supervised, Unsupervised & Other Machine Learning Methods
- Snowflake: Using Analytics & Statistical Functions
- Data Ethics for Companies