



Why Python Simplifies the Use of Machine Learning Models for Advanced Analytics

White Paper

www.indiumsoftware.com



Python is a versatile and mature tool that not only has had a headstart over comparable tools by virtue of the number of years but also because of its quick prototyping capabilities and also because it enables AI development. This makes it a popular choice for machine learning models as well as advanced analytics.

The Python code is similar to English, making it easy to learn, read and write. Being OpenSource, it does not need a license for use. Community support, development packages, libraries and frameworks are available that further speed up the development process due to the

availability of reusable components. It can be made available on any platform, including the cloud.

According to GitHub, globally, Python's popularity was responsible for a near 16.7% growth in the last 5 years, while Java experienced a negative trend of -7.0%!

Though suited for all kinds of developments, Python is also highly scalable and can handle vast volumes of data requests simultaneously. This is yet another reason why Python is suitable for creating machine learning models for advanced analytics.

1. Python for AI/ML

Some of the factors that contribute to its being the most preferred language of development for machine learning in advanced analytics include:

Flexibility:

Developers using Python for Machine Learning can choose OOPS or scripting-based programming which allows them to view quickly without having to recompile the Python code completely. The four options that make Python software lower the chances of errors include imperative, object-oriented, procedural style and functional.

Readability:

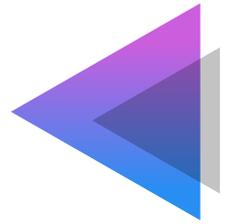
Since it uses the same syntax as English, developers find readability a great boon, making learning and understanding real easy. Even if a developer joins a project mid-way, getting into the thick of things is easy, minimizing the potential for confusion and errors, and enabling swift development of the Machine Learning program.

Visualization:

Analytics requires heavy visualization of data to identify patterns and gain insights from all the variables and factors. Using Python software packages and APIs, it is possible to build histograms, charts, and plots that show the interaction between the different data and how they work together.

Full-Stack Development:

Python lends itself to developing all kinds of applications, both front end and back end. It facilitates communication between computer systems and execute software. It can also interact with code written in other languages such as C and JavaScript.



2. Libraries and Frameworks Aiding AI / ML Development

Machine learning is the training of algorithms to identify data patterns and requires data to be harnessed, refined, processed, cleaned, and sorted out for insights to be derived to create intelligent algorithms. Apart from the simplicity of the language, the fact that the vast community of Python developers has also created libraries that help with different aspects of advanced analytics makes it ideal for the purpose. Some of the key libraries include:

NumPy:

NumPy, a library for multidimensional array objects, facilitates scientific computing to reduce the computational complexity of the Python program.

SciPy:

SciPy is a very popular library that is used for scientific and heavy computing that involves mathematics and engineering. A user-friendly library, it makes available numerical optimization and integration routines.

Scikit-Learn:

Built on NumPy and SciPy for data mining and data analysis, it allows ML engineers to use an integrated approach for AI projects and enables both supervised and unsupervised learning.

In addition to these six, Python also offers advanced libraries and frameworks such as:

TensorFlow:

This enables you to develop and train highly efficient Machine Learning and Deep Learning models, and conduct high-performance numerical and scientific calculations that can be deployed on platforms such as a CPU, GPU, or a TPU (Tensor Processing Unit).

PyTorch:

Launch Machine Learning, Deep Learning, and Neural Networks projects that leverage the high-performance Torch library, for quick flexible development with built-in support for GPU acceleration.

Pandas:

It facilitates EDA (explorative data analysis) and visualization with high-level data structures and several tools for working closely with multiple datasets.

Keras:

An API that runs over TensorFlow, Keras allows developers with quick experimentation for Artificial Intelligence and offers a great user experience.

Matplotlib:

Matplotlib, enables the creation of charts, histograms, scatterplots, and so on using data visualization and exploration and facilitates the customization of Python AI projects.

Theano:

Enables intensive mathematical and scientific calculations to perform complex, multi-dimensional arrays such as defining and evaluating different types of operations on mathematical expressions.

Seaborn:

It allows you to create appealing, informative statistical graphics and facilitates drawing a correlation between multiple variables. It also allows analyzing uni-variate or bi-variate distributions, plot linear regression models for dependent variables and provide high-level abstractions, multi-plot grids.

**Caffe:**

Convolutional Architecture for Fast Feature Embedding, a deep learning framework ideal for scientific projects and industrial implementations, initiates language, speed, and usability. Its solid architecture follows configuration-defined systems that enable optimization without hard coding and helps change between CPU and GPU.

Torch:

A scientific computation system for scientific and numerical processing, it provides a large number of algorithms that are fast, versatile, and usable for increasing performance and promoting deep learning analysis. Cutting and indexing is made simple and includes neural networks and linear algebra protocols.

Django:

A popular, "batteries-included" Python web application framework, the common functionalities for building web applications such as authentication, a template engine, URL routing, database schema migrations and an object-relational mapper (ORM) are integrated with the framework and do not need separate libraries.

Flask:

Developed subsequent to Django, it incorporates all that Python developers seek and coding is more explicit and simple than in Django.

3. Python Best-Suited for MLOps

In the last few years, MLOps, a set of management practices for the deep learning or production ML lifecycle and derived from the terms machine learning or ML and operations or Ops is gaining momentum for the efficient and reliable deployment of ML models in production and maintenance. It promotes greater communication and collaboration between data scientists and operations professionals for the operationalization of ML models.

The need for MLOps emerged to increase the involvement of data scientists in the delivery pipeline of products by drawing the best practices from DevOps. As machine learning and AI demand increased, the traditional way of developing those projects clearly needed a transformation. MLOps enables this by providing the ML engineer with tools to:

- Standardize local development environments.
- Provide parity of development vs. production environments.
- Standardize pipelines for packaging and deployment.

By implementing MLOps, businesses can benefit from:

- Scaling up fast
- Automating repeatable processes to improve productivity, credibility, and reliability,
- Facilitating seamless integration and collaboration between the operations and the ML teams.
- Speeding up production life cycles and promoting better use of data for deriving deeper insights.
- Incorporating best practices in data compliance with regulations such as the Algorithmic Accountability Bill in New York City and the GDPR in the EU.
- Reducing the risk of inaccurate models and avoiding development biases.



Python is well-suited for MLOps too and tools such as Orbyter 1.0 package, with a Python cookiecutter template and a public Docker image, can enable data science teams to adopt Docker for implementing DevOps best practices to improve ML delivery pipelines.

4. Relevant Across Industries, Size of Business

For start-ups, Python provides the flexibility to experiment, develop fast, test fast and fail fast to improve performance and time to market. As a result, Python is a very handy tool that can provide the speed that these businesses are seeking.

For small and medium businesses wishing to leverage the Industry 4.0 technology and data analytics may be a Utopian dream due to the formidable cost of migration. However, by leveraging Python, which can be easy on the pocket, the businesses can not only benefit from digital transformation but Python also promises scalability to help them achieve their roadmap for future growth.

In the case of established enterprises, Python is viable via media that lets them protect their existing investments while leveraging the cloud for advanced analytics and other functions.

The easy availability of resources, the lower cost of development due to Python being OpenSource, the access to packages, libraries and frameworks all make it a delight for developers and business owners alike.

Indium Software, a two-decade-old software development and solutions company, has an array of qualified Python developers with experience in advanced analytics and data science. Our approach to advanced analytics includes

- Interactive data exploration and visualization
- Machine Learning and Model Training
- Real-Time or Batch Predictive Processing

By automating Data Analytics with tools such as Python, Indium enables:

- Reduced operational costs
- Increased operational efficiency
- Enhanced self-service modules
- Increased scalability of big data technologies

Our range of consulting services include:

- Cognitive Analytics
- Big Data Analytics
- Customer Analytics
- Big Data Pricing Analytics
- Pricing Analytics Services
- Social Media Analytics Services



INDIA

Chennai | Bengaluru | Mumbai
Toll-free: 1800 123 1191

USA

Cupertino | Princeton
Toll-free: 1 888 207 5969

UK

London

SINGAPORE

+65 9630 7959



Sales Inquiries

sales@indiumsoftware.com

General Inquiries

info@indiumsoftware.com

