
Article

[Eduard Lebedyuk](#) · Apr 12, 2019 1m read

[Open Exchange](#)

Python Gateway Part I: Introduction

This series of articles would cover [Python Gateway](#) for InterSystems Data Platforms. Leverage modern AI/ML tools and execute Python code and more from InterSystems IRIS. This project brings you the power of Python right into your InterSystems IRIS environment:

- Execute arbitrary Python code
- Seamlessly transfer data from InterSystems IRIS into Python
- Build intelligent Interoperability business processes with Python Interoperability Adapter
- Save, examine, modify and restore Python context from InterSystems IRIS

Index

The plan for the series so far (subject to change).

- Part I: Overview, Landscape and Introduction <-- you're here
- Part II: Installation and Troubleshooting
- Part III: Basic functionality
- Part IV: Interoperability Adapter
- Part V: Execute function
- Part VI: Dynamic Gateway
- Part VII: Proxy Gateway
- Part VIII: Use cases and ML Toolkit

Overview

Machine learning (ML) - is the study of algorithms and statistical models to effectively perform a specific task without using explicit instructions, relying on patterns and inference instead.

Machine learning algorithms and models are becoming more and more commonplace. There is a [variety of reasons](#) for that, but it all comes down to affordability, simplicity and producing actionable results. Is clustering or even neural network modeling a new technology? Of course not, but nowadays you do not need to write hundreds of thousands lines of code to run one and the costs are much more manageable.

Tools are evolving - while we currently do not have completely GUI-based AI/ML tools, but the same progress we saw with many other computer technologies, most notable being BI tools (from writing code to utilizing frameworks to GUI-based configurable solutions) is seen with AI/ML tools. We already passed the point of writing code and are currently utilizing frameworks to configure and calculate the models.

Other improvements, i.e. distributing pre-trained model, where end user should just finish model training on a real-life data also simplify onboarding process. These advances make getting into data science a much easier endeavor for both individuals and companies.

On the other hand nowadays we collect more data about every transaction business makes. With a unified data platform such as InterSystems IRIS all this information can be accessed immediately and used as a fuel for predictive models.

With the other big mover – cloud, running AI/ML workloads becomes easier than ever. Even more important is that we can consume only the resources we require. Moreover, with massive parallelization offered by cloud platforms we can save on a time to a working solution.

But what about results? Here it gets a little trickier. There are lots of tools to build a model, and I ' ll talk about them later, and it ' s not always easy to build a good model, but what comes after? Extracting business value from a model is also a nontrivial endeavor. The root of the problem is the separation of analytical and transactional data flows and data models. When we train the model, we usually do that on a historical data in a warehouse system. But the greatest place for the built model to be is in the heart of transactional processing. What good is the best fraud detection model if we run it once a day? The criminals would be long gone with the money. We need to train the model on a historical data but we also need to apply the model in a real time on the new incoming data so that our business processes can act on predictions the model makes.

MLToolkit

MLToolkit is a comprehensive set of tools, which aims to do exactly that – bring predictive models and transactional environments together, so that the models you build can be easily leveraged right inside your business processes. Titular Python Gateway is a part of MLToolkit and provides integration with a Python language.

Landscape

Before we go further, I would like to describe several tools and libraries for Python, which we would use later.

Tools

- Python is an interpreted, high-level, general-purpose programming language. The main advantage of the language is a big library of mathematical, ML and AI libraries. Same as ObjectScript it's an object-oriented language but everything is dynamic rather than static. Also, everything is an object. The later articles assume a passing familiarity with the language. If you want to start learning, I recommend [starting with documentation](#).
- For our later exercises [install Python 3.6.7 64 bit](#).
- IDE: I use [PyCharm](#), but there are a [lot of them](#). If you're using Atelier, Eclipse for Python developers is a thing.
- Notebook: instead of IDE you can write and share your scripts in a Web-based notebook. The most popular one is [Jupyter](#).

Libraries

Here's a (incomplete) list of libraries used for Machine Learning.

- [Numpy](#) is the fundamental package for scientific computing with Python.
- [Pandas](#) is a library providing high-performance, easy-to-use data structures and data analysis tools.
- [Matplotlib](#) is a 2D plotting library which produces figures in a variety of hardcopy formats and interactive environments across platforms.
- [Seaborn](#) is a data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.
- [Sklearn](#) is a machine Learning library.
- [XGBoost](#) is an optimized distributed gradient boosting library designed to be highly efficient, flexible and portable. It implements machine learning algorithms under the Gradient Boosting framework.
- [Gensim](#) is a library for unsupervised topic modeling and natural language processing.
- [Keras](#) is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano.
- [Tensorflow](#) is an end-to-end open source machine learning platform.

- [PyTorch](#) deep learning platform similar to Tensorflow but Python focused.
- [Nyoka](#) produces PMML from Python models.

Summary

AI/ML technologies allow business to be more effective and more adaptable. Moreover, today these technologies are becoming easier to build and deploy. Start investigating AI/ML technologies and how it can help your organization to grow and prosper. There are examples, stories and use cases from almost every industry. Do not miss your chance to use future technologies today.

What's next

In the next part we would install Python Gateway. Don't forget to register for the upcoming webinar (details below)!

Links

- [Python Gateway](#)
- [Install Python 3.6.7 64 bit](#)
- [Python documentation/tutorial](#)

Webinar

Do you want to reap the benefits of the advances in the fields of artificial intelligence and machine learning? With InterSystems IRIS and the Machine Learning (ML) Toolkit, it ' s easier than ever. Join my colleague Sergey Lukyanchikov and me on Tuesday, April 23rd at 11am EDT for the Machine Learning Toolkit for InterSystems IRIS webinar to find out how InterSystems IRIS can be used as both a standalone development platform and an orchestration platform for AI/ML models that brings together InterSystems IRIS, Python and other external tools.

Date: Tuesday, April 23rd at 11am EDT

Recommended Audience: Developers, Solution Architects, Data Scientists, and Data Engineers.

[REGISTER NOW!](#)

[#AI](#) [#Best Practices](#) [#Languages](#) [#Machine Learning](#) [#Python](#) [#InterSystems IRIS](#)
[Check the related application on InterSystems Open Exchange](#)

Source URL: <https://community.intersystems.com/post/python-gateway-part-i-introduction>