Artículo

[Pierre-Yves Duq...](#) · 7 mayo, 2020 · Lectura de 6 min

Python Gateway. Parte I: Introducción

Esta serie de artículos describe el uso del [Python Gateway](#) para InterSystems IRIS. Python Gateway permite acceder a toda la potencia de las librerías Python y las herramientas de Aprendizaje máquina (IAML) desde InterSystems IRIS y:

- Ejecutar cualquier código Python
- Transferir datos de forma transparente desde IRIS a Python
- Construir procesos de interoperabilidad inteligente con el Adaptador de Python
- Guardar, revisar, modificar y restaurar el contexto de Python desde InterSystems IRIS

Índice

El plan para la serie hasta el momento (sujeto a cambios).

- Parte I: Resumen, panorama e introducción <-- usted está aquí
- Parte II: Instalación y resolución de problemas
- Parte III: Funcionalidades básicas
- Parte IV: Adaptador de Interoperabilidad
- Parte V: Cómo ejecutar una función
- Parte VI: Puerta de enlace dinámica
- Parte VII: Puerta de enlace proxy
- Parte VIII: Casos de usos y conjunto de herramientas de aprendizaje automático

Resumen

El aprendizaje automático (ML, por el inglés "Machine Learning") es el estudio de algoritmos y modelos estadísticos para realizar de forma efectiva una tarea específica sin usar instrucciones específicas, confiando en vez en patrones e inferencias.

Los algoritmos y modelos de aprendizaje automático son cada vez más utilizados. Hay una [variedad de motivos](#) para eso, pero en el fondo pasa por la economía, la simplicidad y la creación de resultados útiles. ¿El clustering (análisis de grupos) o incluso el modelado de redes neuronales son nuevas tecnologías? Por supuesto que no, pero hoy en día ya no es necesario escribir cientos de líneas de código para ejecutar uno.

Las herramientas están evolucionando. Si bien aún no tenemos herramientas de IA/ML totalmente basadas en una interfaz gráfica, observamos en este caso el mismo desarrollo que vimos con muchas otras tecnologías informáticas, en particular las herramientas de Inteligencia empresarial (BI) (desde escribir código a utilizar frameworks a soluciones configurables basadas en interfaces gráficas). Ya pasamos el punto de escribir código y actualmente usamos frameworks para configurar y calcular los modelos.

Otras mejoras, como uso de modelos pre-entrenados (" transfer learning " en inglés) , en el que el usuario final sólo debe terminar el entrenamiento de un modelo genérico con sus datos propios más específicos,

también simplifican el proceso de adaptación. Estos avances hacen empezar en ML sea mucho más asequible tanto para individuos como para empresas.

Por otro lado, hoy en día recolectamos más datos sobre cada transacción que realiza una empresa. Con una plataforma de datos unificada, como InterSystems IRIS, es posible acceder a toda esta información de forma inmediata para usarla como alimento para los modelos predictivos.

Gracias al otro gran motor, la nube, procesar cargas de trabajo de IA/ML se vuelve más fácil que nunca. Permite consumir los recursos necesarios sin costes de adquisición, y gracias a la paralelización masiva que ofrecen las plataformas de nube, podemos ahorrar tiempo de implementación de una solución.

¿Pero qué pasa con los resultados? Aquí hay cierta complejidad. Existen muchas herramientas para construir un modelo, de las cuales hablaré más adelante, y no siempre es fácil construir un buen modelo, ¿pero qué viene después? Extraer valor comercial de un modelo tampoco es una tarea trivial. La raíz del problema es la separación de flujos de datos analíticos y transaccionales. Cuando entrenamos al modelo, generalmente lo hacemos con datos históricos de un repositorio analítico. Pero el mejor lugar para usar este modelo es en el corazón del procesamiento transaccional. ¿De qué sirve el mejor modelo de detección de fraude si lo ejecutamos una vez por día? Las transacciones fraudulentas ya habrán sido aceptadas y será demasiado tarde para tomar acción. Necesitamos entrenar un modelo en base a datos históricos, pero también necesitamos aplicar el modelo en tiempo real sobre los nuevos datos entrantes, para que nuestro proceso empresarial pueda actuar en base a las predicciones que realiza el modelo.

MLToolkit

MLToolkit es un completo conjunto de herramientas que apunta a hacer exactamente eso: unir los modelos predictivos a los entornos transaccionales, para que pueda aprovechar los modelos que construya directamente dentro de su proceso empresarial. El Python Gateway es parte del MLToolkit y brinda integración con un lenguaje Python.

Panorama

Antes de continuar, quisiera describir varias herramientas y bibliotecas para Python, que usaremos más tarde.

Herramientas

- Python es un lenguaje de programación interpretado, de propósito general y de alto nivel. La principal ventaja del lenguaje es el gran número de librerías matemáticas, de aprendizaje automático e inteligencia artificial. Al igual que ObjectScript, se trata de un lenguaje orientado a objetos y dinámico. Los siguientes artículos asumen que el lector tiene una familiaridad básica con el lenguaje. Si desea comenzar a aprender, le recomiendo [comenzar con la documentación](#).
- Para nuestros siguientes ejercicios [instale Python 3.6.7 64 bit](#).
- IDE: Yo uso [PyCharm](#), apuesto a que hay [muchos otros](#). Si usa Atelier, existe Eclipse para desarrolladores Python.
- Bloc de notas: en lugar del IDE, puede escribir y compartir sus scripts en un bloc de notas web. El más popular es [Jupyter](#).

Bibliotecas

A continuación encontrará una lista (incompleta) de bibliotecas usadas para aprendizaje automático.

- [Numpy](#) es el paquete fundamental para programación científica con Python.
- [Pandas](#) es una biblioteca que brinda estructuras de datos y herramientas de análisis de datos que son fáciles de usar y potentes.
- [Matplotlib](#) es una biblioteca gráfica 2D para figuras en varios formatos del entorno o impresos.
- [Seaborn](#) es una biblioteca de visualización de datos basada en matplotlib. Brinda una interfaz de alto nivel para crear gráficos estadísticos atractivos e informativos.
- [Sklearn](#) es una biblioteca para aprendizaje automático.
- [XGBoost](#) es una biblioteca optimizada para mejorar los gradientes distribuidos, diseñada para tener ser altamente eficiente, flexible y portátil. Implementa algoritmos de aprendizaje automático bajo el framework Gradient Boosting.
- [Gensim](#) es una biblioteca para el modelado de temas no supervisado y procesamiento de lenguaje natural.
- [Keras](#) es una API de alto nivel para redes neuronales, escrito en Python y capaz de ejecutarse sobre TensorFlow, CNTK o Theano.
- [Tensorflow](#) es una plataforma de aprendizaje automático de extremo a extremo de código abierto.
- [PyTorch](#) es una plataforma de aprendizaje profundo similar a Tensorflow pero enfocada en Python.
- [Nyoka](#) genera PMML a partir de modelos de Python.

Resumen

Las tecnologías de inteligencia artificial y aprendizaje automático permiten a las empresas ser más efectivas y más adaptables. Hoy esas tecnologías se están volviendo más fáciles de usar e implementar. Comience a investigar sobre tecnologías de AI/ML y cómo puede ayudar a su organización a crecer y prosperar. Aquí hay ejemplos, historias y casos de uso para casi cualquier industria. No se pierda la oportunidad de usar hoy las tecnologías del futuro.

¿Cómo seguir?

En la siguiente sección, instalaremos el Python Gateway. ¡No olvide registrarse para el próximo seminario web (detalles a continuación)!

Enlaces

- [Python Gateway](#)
- [Instalar Python 3.6.7 64 bits](#)
- [Documentación/tutorial Python](#)

[#Análítica convergente](#) [#Inteligencia Artificial](#) [#Machine learning](#) [#Python](#) [#InterSystems IRIS](#)

URL de fuente: <https://es.community.intersystems.com/post/python-gateway-parte-i-introducci%C3%B3n>