MLOps / ML model deployment

ML model deployment

ubuntu.com/blog/guide-to-ml-model-serving

CANONICAL

ubuntu® Enterprise Developer Community Download

Blog Overview Internet of Things Desktop Cloud and Server Topics

A guide to ML model serving

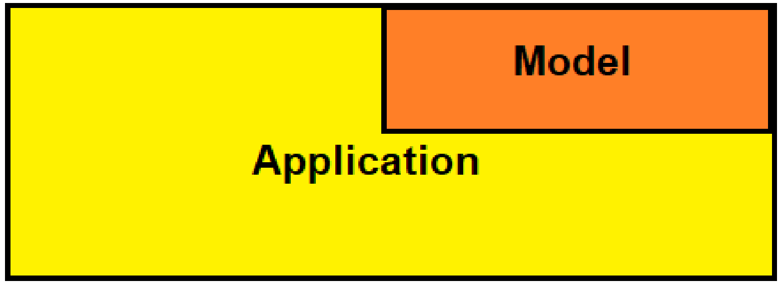
Rui Vasconcelos

on 17 May 2021

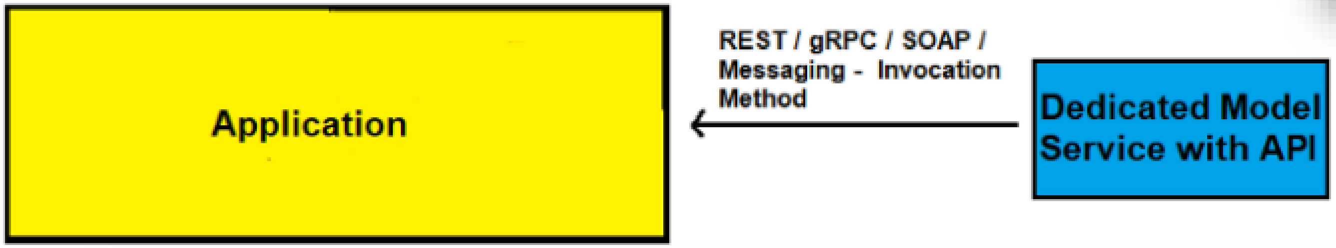
Tags: AI/ML , deep learning , Kubeflow , kubernetes , machine learning

> TL;DR: How you deploy models into production is what separates an academic exercis from an investment in ML that is value-generating for your business. At scale, this becomes painfully complex. This guide walks you through industry best practices and methods, concluding with a practical tool, KFServing, that tackles model serving at scale.

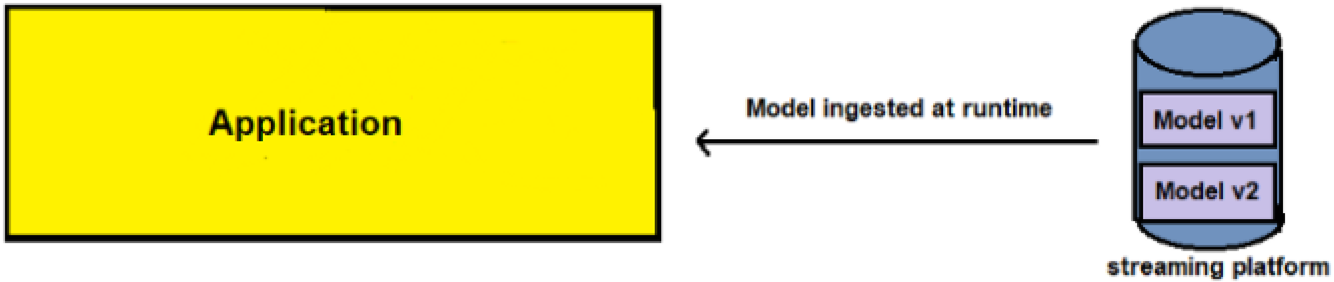
#1 Model Embedded in Application



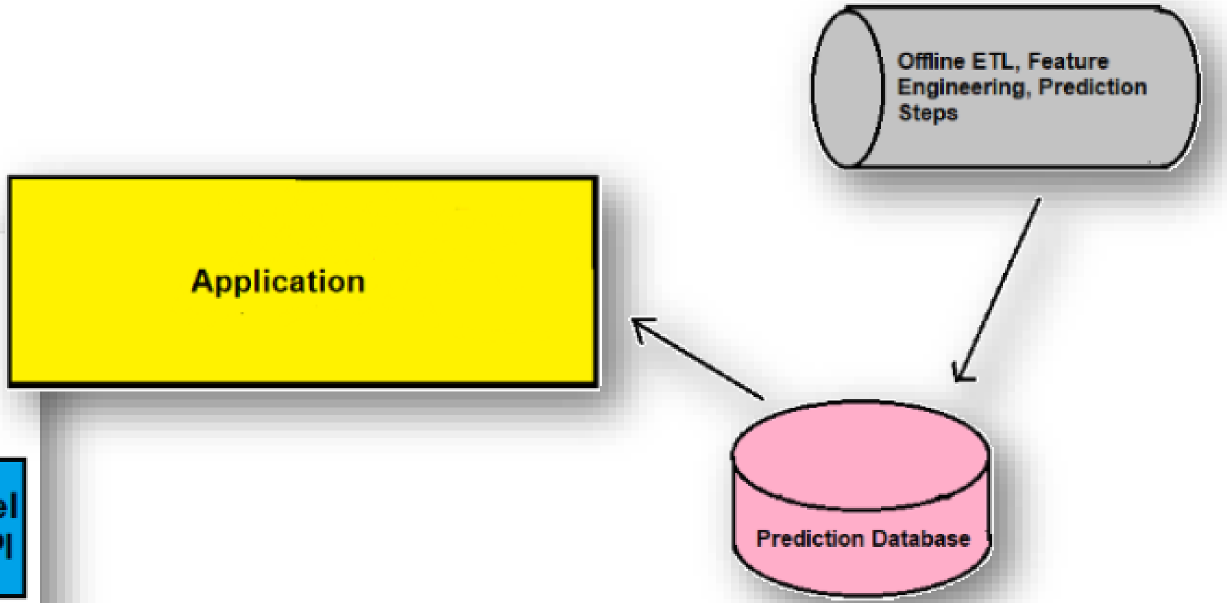
#2 Dedicated Model API



#3 Model Published as Data



#4 Offline Predictions



https://towardsdatascience.com/4-machine-learning-system-architectures-e65e33481970

ML deployment approaches

- **Official standards like Open Neural Network Exchange (ONNX), Portable Format for Analytics (PFA) or Predictive Model Markup Language (PMML):** A data scientist builds a model with Python. The Java developer imports it in Java for production deployment. This approach supports different frameworks, products and cloud services. You do not have to rely on the same framework or product for training and model deployment. Consider ONNX, a relatively new standard for deep learning-it already supports TensorFlow, PyTorch and MXNet. These standards have pros and cons. Some people like and use them; many don't.
- **Developer-focused frameworks like Deeplearning4j:** These frameworks are built for software engineers to build the whole machine learning lifecycle on the Java platform, not just model deployment and monitoring but also preprocessing and training. You can still import other models if you want (e.g., Deeplearning4j lets you import Keras models). This option is great if you: a) have data scientists who can write Java or b) have software engineers who understand machine learning concepts enough to build analytic models.
- **AutoML for building analytic models with limited machine learning experience:** This way, domain experts can build and deploy analytic models with a button click. The AutoML engine provides an interface for others to use the model for predictions.
- **Embedding model binaries into applications:** The output of model training is an analytic model. Fo instance, you can write Python code to train and generate a TensorFlow model. Depending on the framework, the output can be text files, Java source code or binary files. For example, TensorFlow generates a model artifact with Protobuf, JSON and other files. No matter what format the output of your machine learning framework is, it can be embedded into applications to use for predictions via the framework's API (e.g., you can load a TensorFlow model from a Java application through TensorFlow's Java API).
- **Managed model server in the public cloud like Google Cloud Machine Learning Engine:** The cloud provider takes over the burden of availability and reliability. The data scientist "just" deploys its

Model Serving and Monitoring

- Backprop Stars 192 - Backprop makes it simple to use, finetune, and deploy state-of-the-art ML models.
- BentoML Stars 2.6k - BentoML is an open source framework for high performance ML model serving
- Cortex Stars 7.5k - Cortex is an open source platform for deploying machine learning models—trained with any framework—as production web services. No DevOps required.
- DeepDetect Stars 6 - Machine Learning production server for TensorFlow, XGBoost and Cafe models written in C++ and maintained by Jolibrain
- Evidently Stars 713 - Evidently helps analyze machine learning models during development, validation, or production monitoring. The tool generates interactive reports from pandas DataFrame.
- ForestFlow Stars 47 - Cloud-native machine learning model server.
- Jina Stars 7.3k - Cloud native search framework that supports to use deep learning/state of the art AI models for search.
- KFServing Stars 972 - Serverless framework to deploy and monitor machine learning models in Kubernetes - (Video)
- Model Server for Apache MXNet (MMS) Stars 806 - A model server for Apache MXNet from Amazon Web Services that is able to run MXNet models as well as Gluon models (Amazon's SageMaker runs a custom version of MMS under the hood)
- OpenScoring Stars 546 - REST web service for scoring PMML models built and maintained by OpenScoring.io
- Redis-AI Stars 591 - A Redis module for serving tensors and executing deep learning models. Expect changes in the API and internals.
- Seldon Core Stars 2.4k - Open source platform for deploying and monitoring machine learning models in kubernetes - (Video)
- Tempo Stars 45 - Open source SDK that provides a unified interface to multiple MLOps projects that enable data scientists to deploy and productionise machine learning systems.
- Tensorflow Serving Stars 5.1k - High-performant framework to serve Tensorflow models via grpc protocol able

aws.amazon.com/blogs/aws/amazon-sagemaker-edge-manager-si

Amazon SageMaker Edge Manager

Manage and monitor ML models efficiently across fleets of smart devices

aws

Products Solutions Pricing Documentation Learn Partner Network AWS Marketplace Customer Enablemer

Blog Home Category Edition Follow

AWS News Blog

Amazon SageMaker Edge Manager Simplifies Operating Machine Learning M

by Julien Simon | on 08

Re:Invent, Events, Inter

0:00 / 0:00

Today, I'm extremely
makes it easier to op

Edge computing is c
continued advances
numbers of embedd
agriculture, healthca
purpose: capture da

As machine learning
to edge applications
from local data. Ho

Model management across fleets of edge devices

Optimize ML models for a wide range of devices

Amazon SageMaker Edge Manager automatically optimizes ML models for deployment on a wide variety of edge devices, including devices powered by CPUs, GPUs, and embedded ML accelerators. SageMaker Edge Manager compiles your trained model into an executable that discovers and applies specific performance optimizations that can make your model run up to 25x faster on the target hardware. SageMaker Edge Manager allows you to optimize and package trained models using different frameworks such as DarkNet, Keras, MXNet, PyTorch, TensorFlow, TensorFlow-Lite, ONNX, and XGBoost for inference on Android, iOS, Linux, and Windows based machines.

Easy integration with device applications

Amazon SageMaker Edge Manager supports gRPC, an open source remote procedure call, which allows you to integrate SageMaker Edge Manager with your existing edge applications through APIs in common programming languages, such as Android Java, C# / .NET, Dart, Go, Java, Kotlin/JVM, Node.js, Objective-C, PHP, Python, Ruby, and Web.

Continuous model monitoring

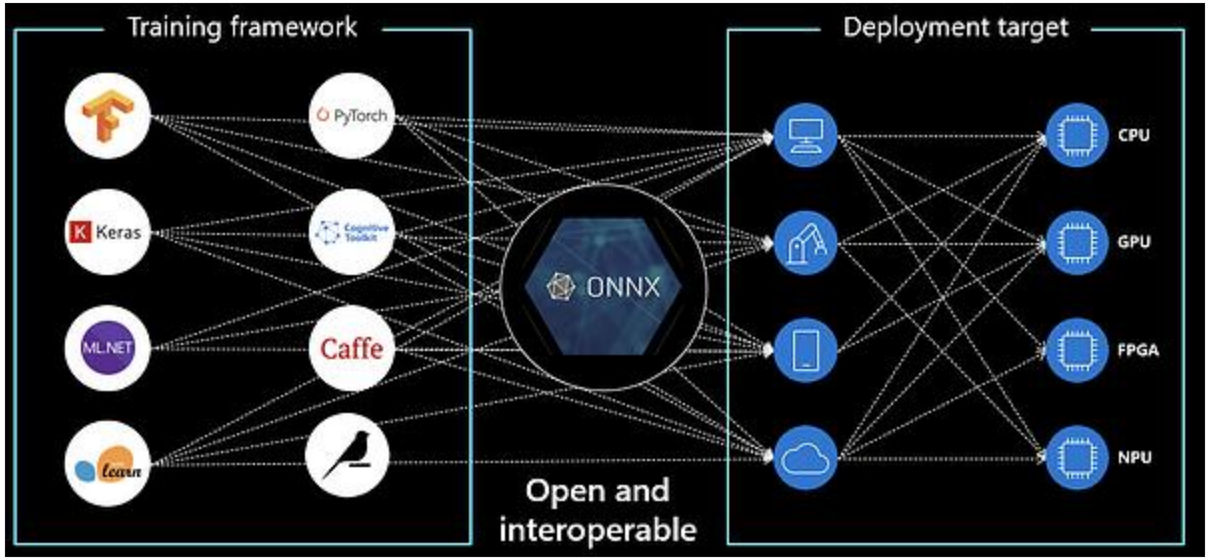
Amazon SageMaker Edge Manager collects data from edge devices and sends a sample to the cloud where it is analyzed and visualized in SageMaker. If quality declines are detected, you can quickly spot them in the dashboard and also configure alerts through Amazon CloudWatch. Declines in model quality, or model drift, can be caused by differences in the data used to make predictions compared to the data used to train the model or by changes in the real world. For example, an object detection model that is not trained on images in snow conditions

Machine learning at the edge: TinyML is getting big

Being able to deploy machine learning applications at the edge is the key to unlocking a multi-billion dollar market. TinyML is the art and science of producing machine learning models frugal enough to work at the edge, and it's seeing rapid growth.

Deploying on the Edge With ONNX

August 19, 2020



ONNX and PMML

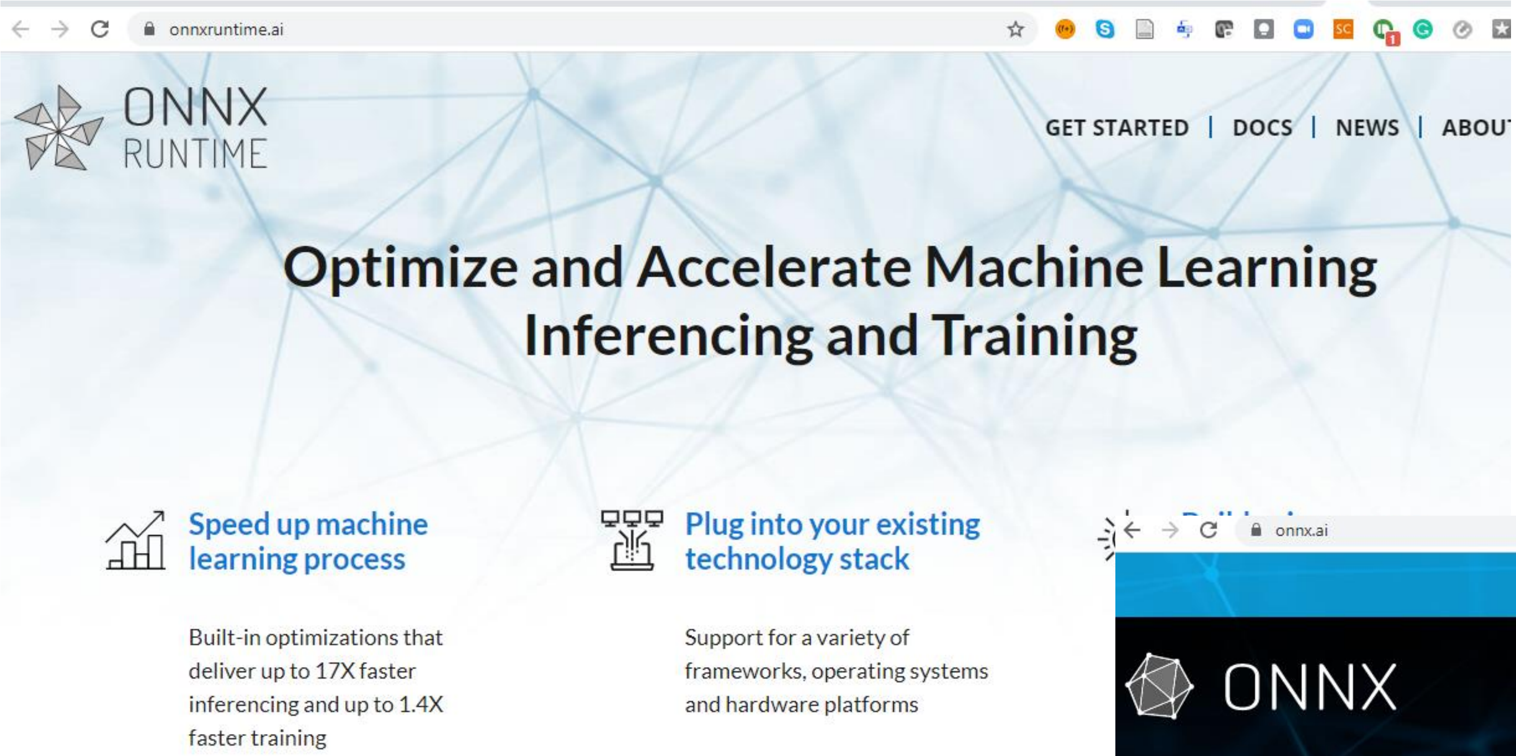
Both ONNX and PMML are model representation standards that have nothing to do with the platform and the environment, which can make model deployment out of the model training environment, simplify the deployment process, and accelerate the rapid launch of the model into the production environment. These two standards have been supported by major manufacturers and frameworks, and have a wide range of applications.

- PMML is a relatively mature standard. Before the birth of ONNX, it can be said to be the actual standard for model representation. It has rich support for traditional data mining models. The latestPMML4.4 Can support up to 19 model types. However, PMML currently lacks support for deep learning models. The next version 5.0 may add support for deep neural networks, but because PMML is based on the old XML format, using text format to store deep neural network model structure and parameters will This brings about the problem of model size and performance. At present, there is no perfect solution to this problem. For a detailed introduction to PMML, you can refer to the article"Deploying Machine Learning Models Using PMML"。

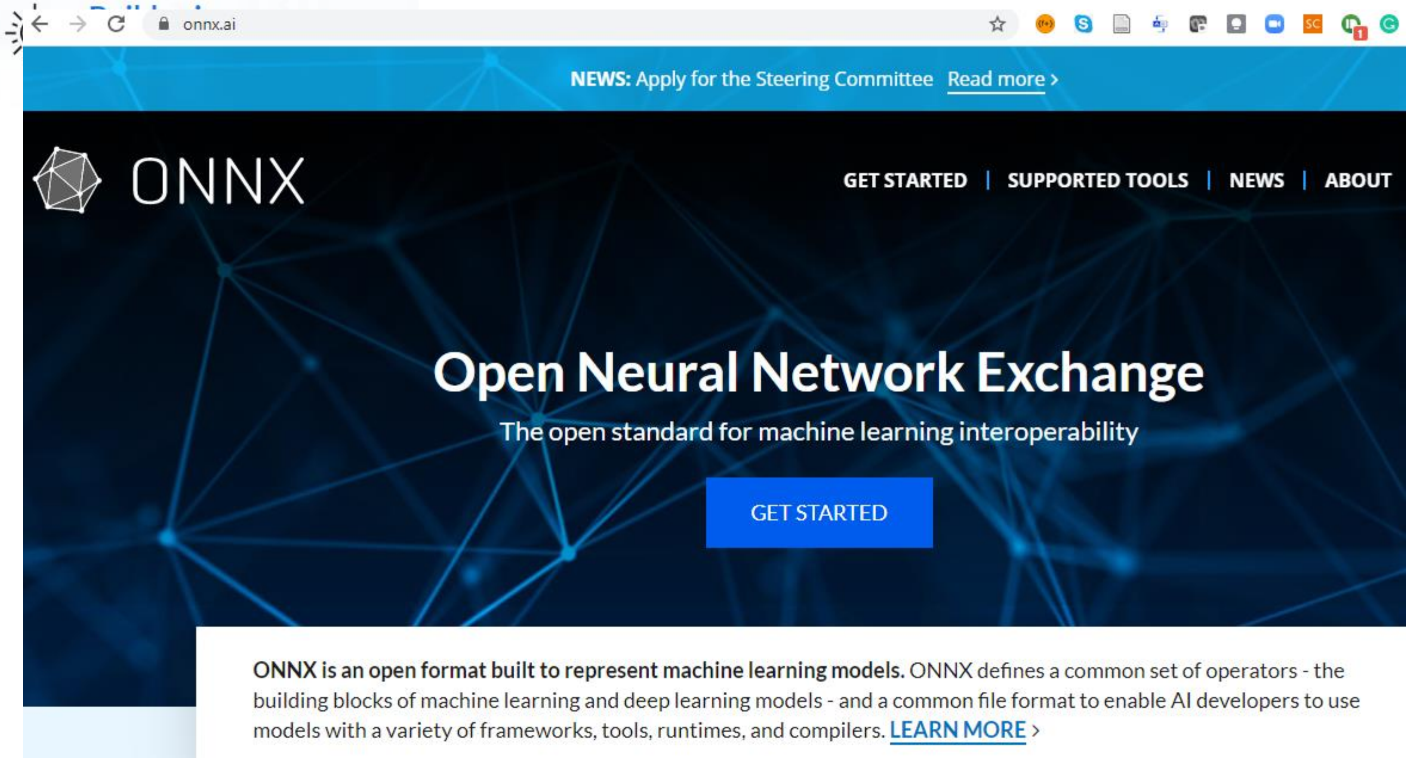
- As a new standard, ONNX initially mainly provided support for deep neural network models to solve the problem of interoperability and exchange of models under different frameworks. Currently passed `ONNX-ML`, ONNX can already support traditional non-neural network machine learning models, but the current model types are not rich enough. ONNX uses the protobuf binary format to serialize the model, which can provide better transmission performance.

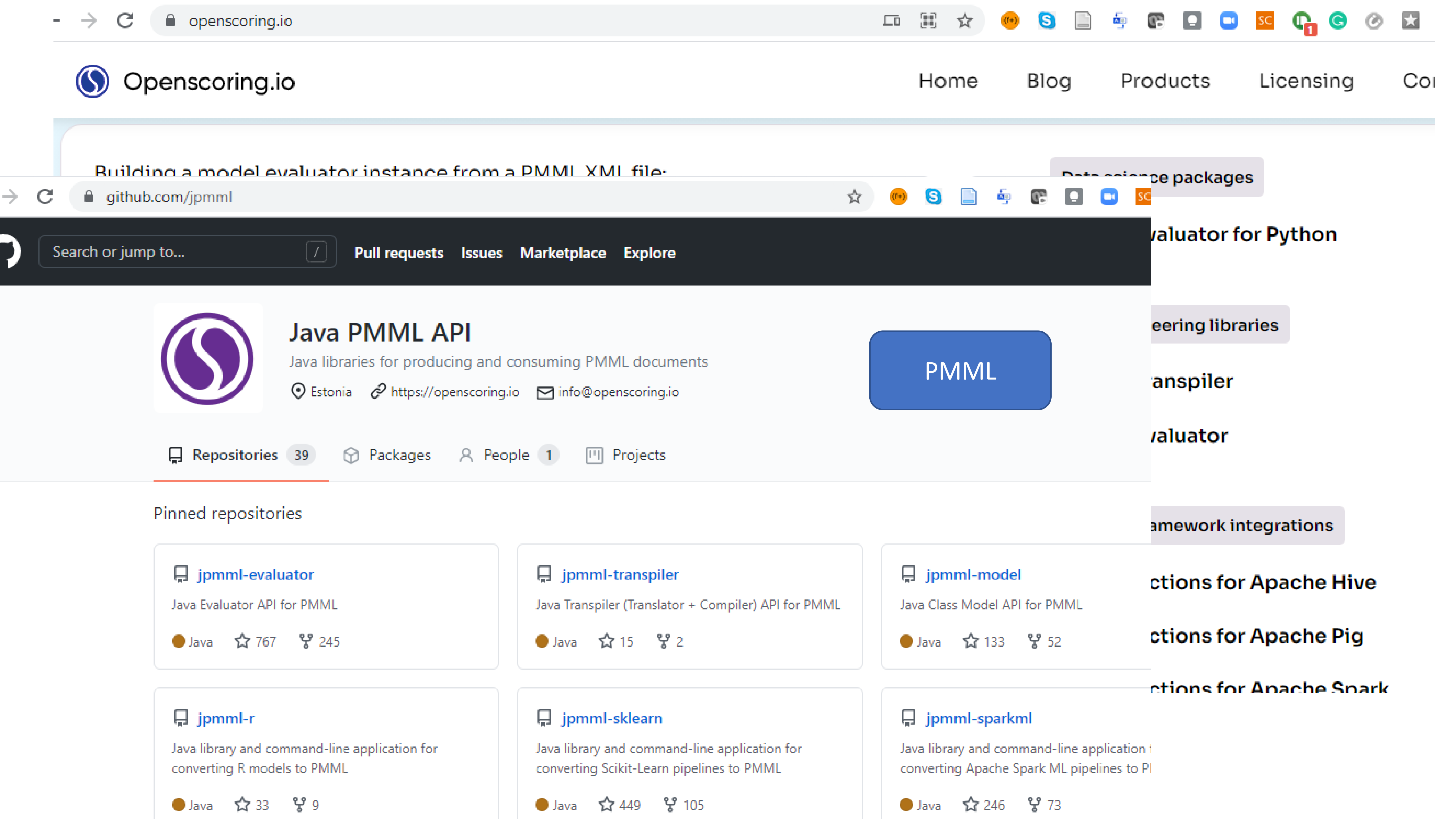
Both ONNX and PMML formats have mature open source libraries and framework support. PMML includes JPMML, PMML4S, PyPMML, etc. ONNX has Microsoft's ONNX runtime, NVIDIA TensorRT and so on. Users can choose the appropriate cross-platform format to deploy the AI model according to their actual situation.

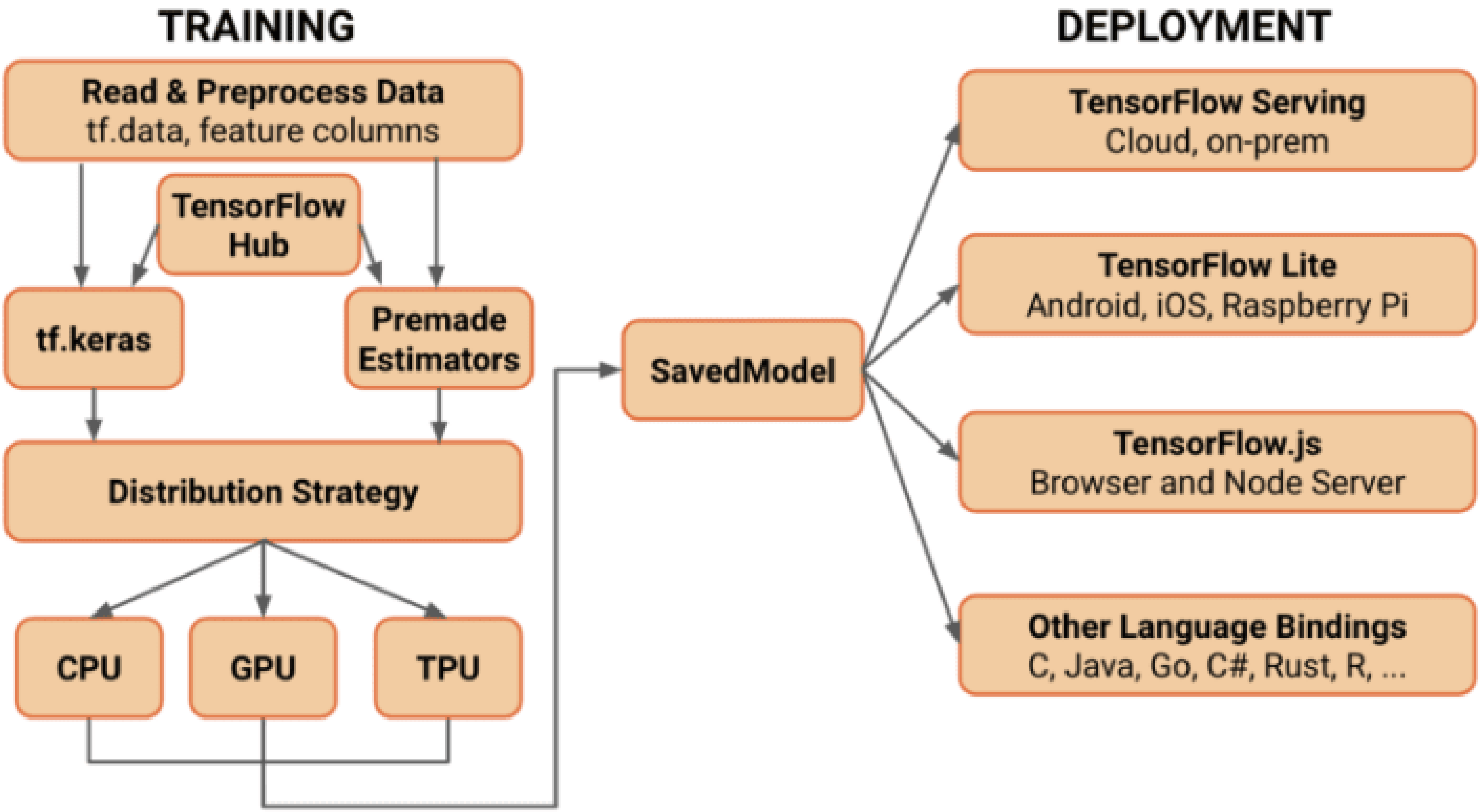
Original ONNX standard



Micrisoft ONNX standard




opescoring.io
Openscoring.io
Home
Blog
Products
Licensing
Building a model evaluator instance from a PMML XML file:
ce packages
valuator for Python
eering libraries
anspiler
valuator
amework integrations
ctions for Apache Hive
ctions for Apache Pig
ctions for Apache Spark
github.com/jpmml
Search or jump to...
Pull requests
Issues
Marketplace
Explore
Java PMML API
Java libraries for producing and consuming PMML documents
Estonia
https://openscoring.io
info@openscoring.io
PMML
Repositories 39
Packages
People 1
Projects
Pinned repositories
jpmml-evaluator
Java Evaluator API for PMML
Java 767 245
jpmml-transpiler
Java Transpiler (Translator + Compiler) API for PMML
Java 15 2
jpmml-model
Java Class Model API for PMML
Java 133 52
jpmml-r
Java library and command-line application for converting R models to PMML
Java 33 9
jpmml-sklearn
Java library and command-line application for converting Scikit-Learn pipelines to PMML
Java 449 105
jpmml-sparkml
Java library and command-line application
converting Apache Spark ML pipelines to P
Java 246 73

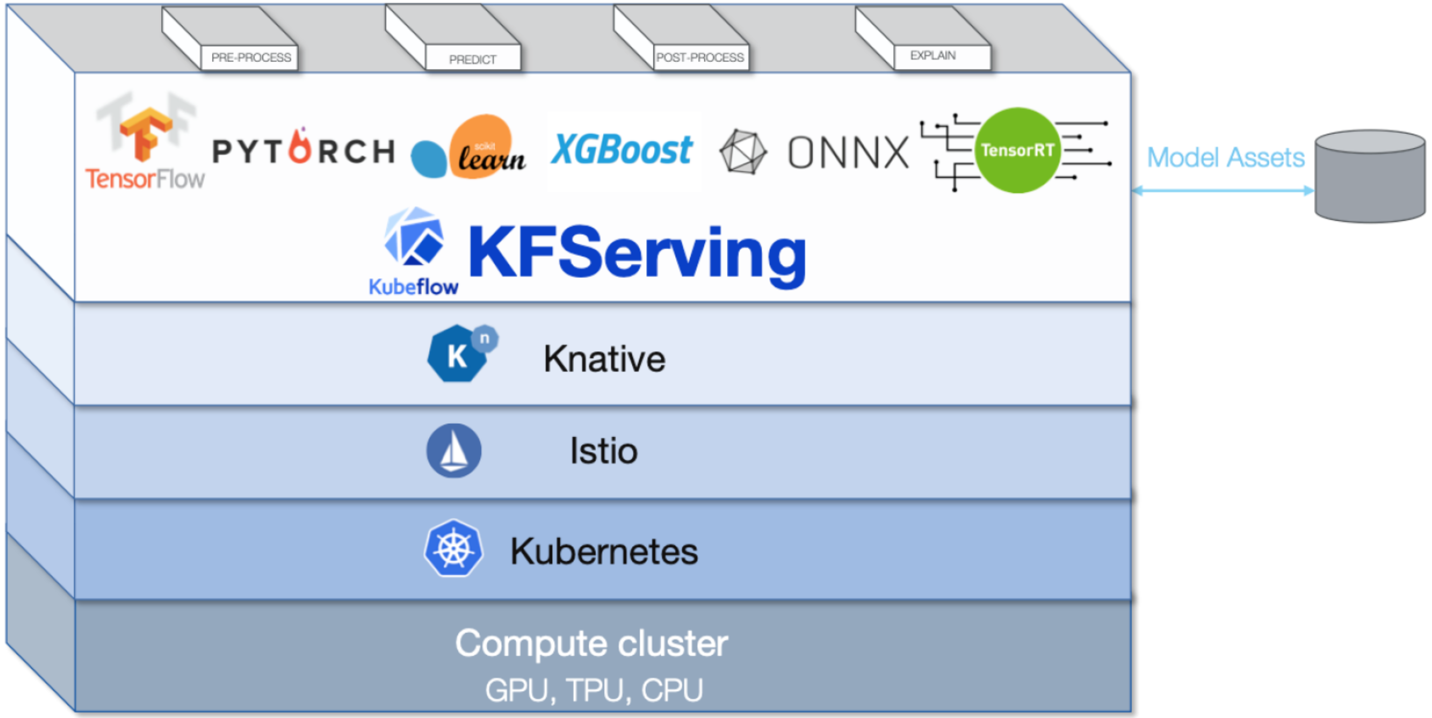

TRAINING
Read & Preprocess Data
tf.data, feature columns
TensorFlow Hub
tf.keras
Premade Estimators
Distribution Strategy
CPU
GPU
TPU
SavedModel
DEPLOYMENT
TensorFlow Serving
Cloud, on-prem
TensorFlow Lite
Android, iOS, Raspberry Pi
TensorFlow.js
Browser and Node Server
Other Language Bindings
C, Java, Go, C#, Rust, R, ...

Serving models on Kubernetes

Enterprise computing is moving to Kubernetes, and Kubeflow has long been talked abou as the platform to solve MLOps at scale.

KFServing, the model serving project under Kubeflow, has shown to be the most mature tool when it comes to open-source model deployment tooling on K8s, with features like canary rollouts, multi-framework serverless inferencing and model explainability.

Learn more about KFServing in What is KFServing?



mlflow

An open source platform for the machine learning lifecycle

Integrations with:

TensorFlow, PyTorch, Keras, Apache Spark, scikit learn, RAPIDS

H2O.ai, python, R, Java, mleap, ONNX

GLUON, XGBoost, LightGBM, spaCy, fast.ai, statsmodels

CatBoost, PyCaret, ALGORITHMIA, OPTUNA, RAY, CONDA

kubernetes, docker, Amazon SageMaker, Azure Machine Learning, Google Cloud, databricks

Streaming Machine Learning with Kafka-native Model Deployment

6 minute read

By  KAI WAEHNER · 27. October 2020

https://docs.seldon.io/projects/seldon-core/en/latest/streaming/kafka.html

SELDON Seldon Core Documentation Search

seldon-core documentation 🚀 Our Other Projects & Products: Alibi Explain Alibi Detect MLServer Tempo SDK Seldon Deplo

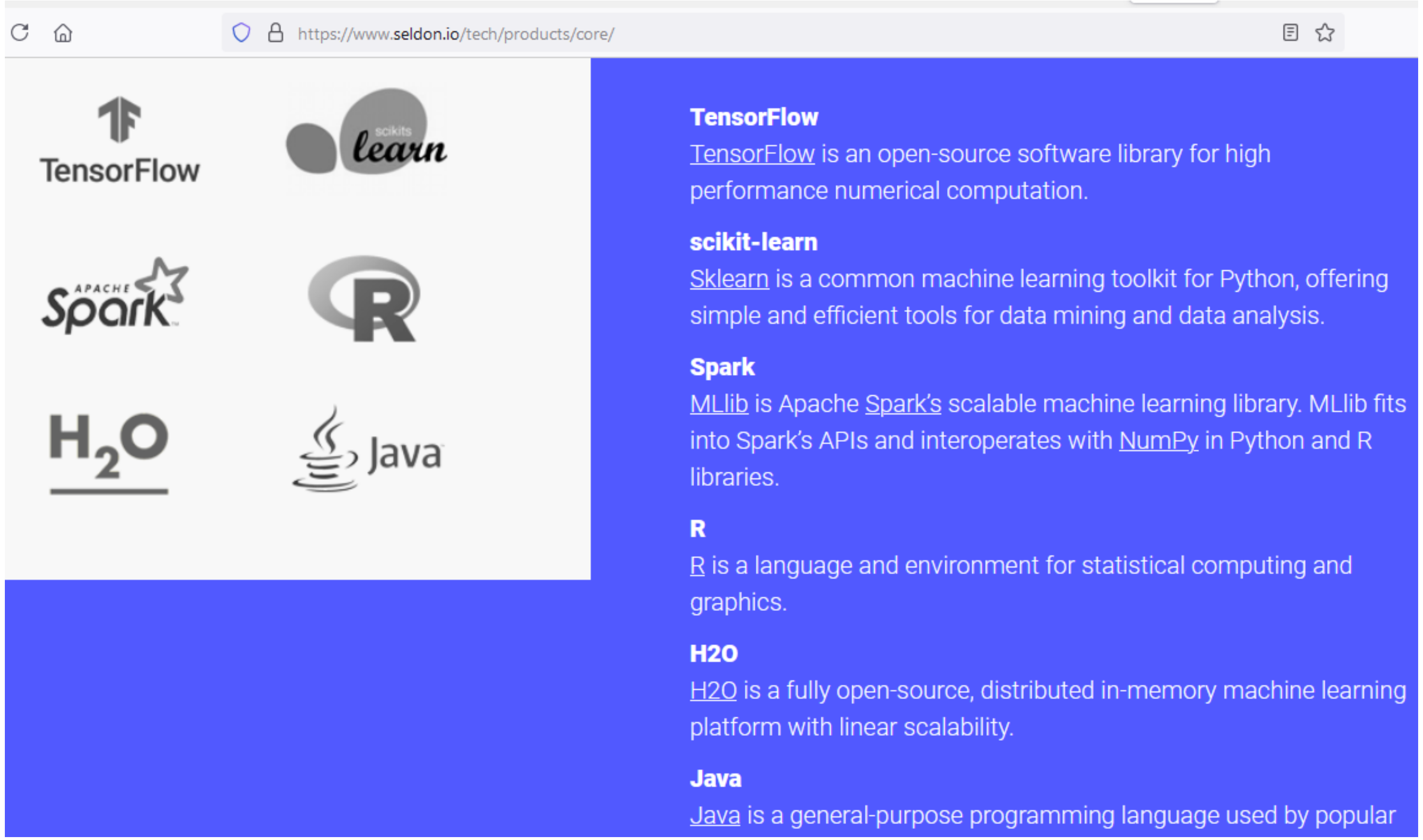
Native Kafka Stream Processing (>=1.2.0, alpha)

Seldon provides a native kafka integration from version 1.2. when you specify `serverType: k` SeldonDeployment.

When `serverType: kafka` is specified you need to also specify environment variables in `svc(` KAFKA_BROKER, KAFKA_INPUT_TOPIC, KAFKA_OUTPUT_TOPIC. An example is shown belo Tensorflow CIFAR10 model:

```
apiVersion: machinelearning.seldon.io/v1
kind: SeldonDeployment
metadata:
  name: tfserving-cifar10
spec:
  protocol: tensorflow
  transport: rest
  serverType: kafka
  predictors:
  - componentSpecs:
    - spec:
        containers:
```

Seldon



https://www.seldon.io/tech/products/core/

TensorFlow
TensorFlow is an open-source software library for high performance numerical computation.

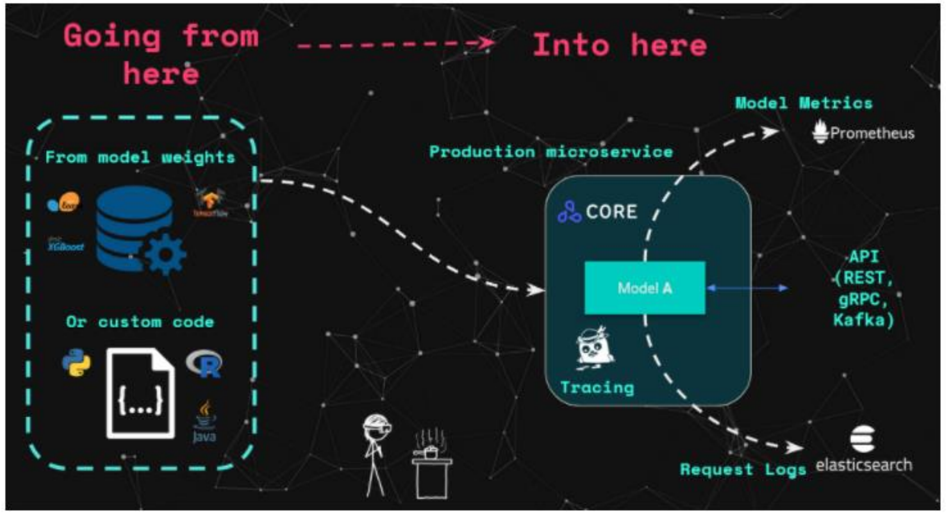
scikit-learn
Sklearn is a common machine learning toolkit for Python, offering simple and efficient tools for data mining and data analysis.

Spark
MLlib is Apache Spark's scalable machine learning library. MLlib fits into Spark's APIs and interoperates with NumPy in Python and R libraries.

R
R is a language and environment for statistical computing and graphics.

H2O
H2O is a fully open-source, distributed in-memory machine learning platform with linear scalability.

Java
Java is a general-purpose programming language used by popular



Real Time Machine Learning at Scale using SpaCy, Kafka & Seldon Core https://hydrosphere.io/product/serving/

hydrosphere.io

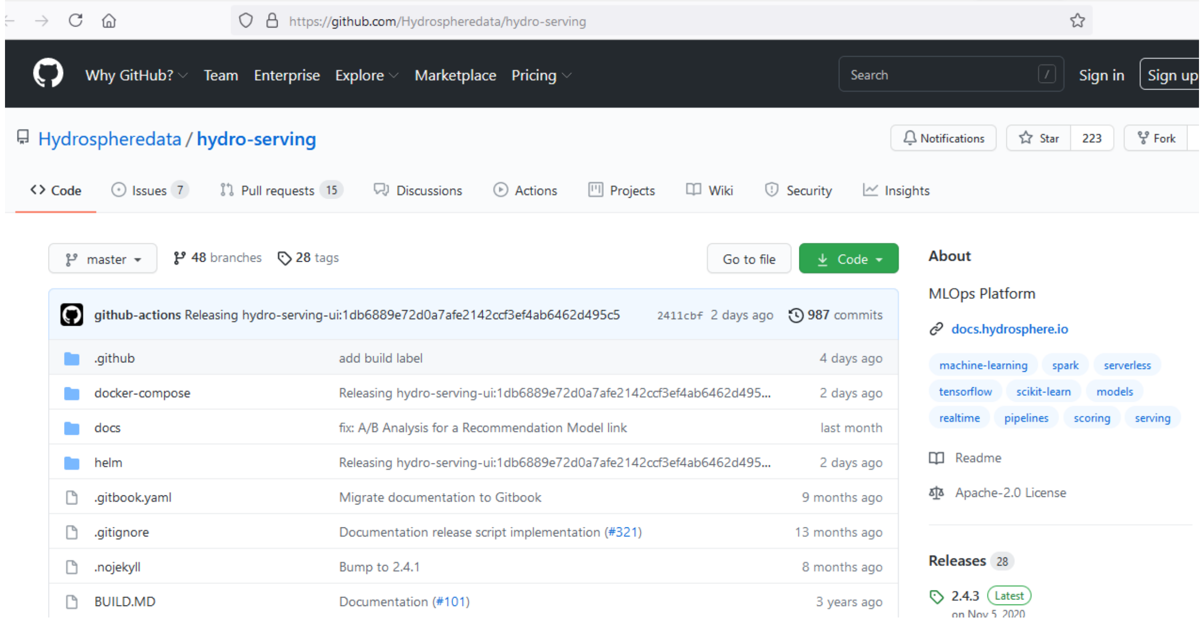
PRODUCT ABOUT US DOCS BLOG REQUEST DEMO

Hydrosphere Serving

Hydrosphere Serving is an open-source cluster for deploying your machine learning models in production. It is a collection of dockerized services that can run anywhere you can run Docker or Kubernetes – any cloud or on-premises.

Features:

- Language- & Framework-agnostic Deployment. No matter which programming language or libraries were used to develop or deploy a model, you still can use Hydrosphere. Python, R, Julia, Scala Spark, custom binary, TensorFlow, PyTorch, etc. are all supported.
- Rich Interfaces. Hydrosphere Serving automatically exposes HTTP, GRPC and Kafka interfaces for your served models.
- Open-Source – enjoy the support of our contributors.
- Model Version Control. Version control your models and pipelines as they are deployed. Explore how metrics change between different model versions and roll-back to a previous version if needed.



Jan 11, 2021

Secure machine learning APIs with Seldon and BentoML

By Steven Reitsma • ML • Istio • Security

BentoML

BentoML is a framework for building and packaging machine learning APIs. When building APIs, many data scientists just `pip install flask`, don't bother with authentication, use the Flask development server and wham, "we're running production". BentoML tackles a lot of these issues:

- It forces you to apply some kind of structure to your APIs, and subsequently exposes a Swagger UI for your API.
- It enables you to easily package your model into a Docker container.
- You can use it to version your machine learning models.
- It uses the highly optimized Gunicorn to serve your application, which is much more performant than the single-threaded Flask development server.

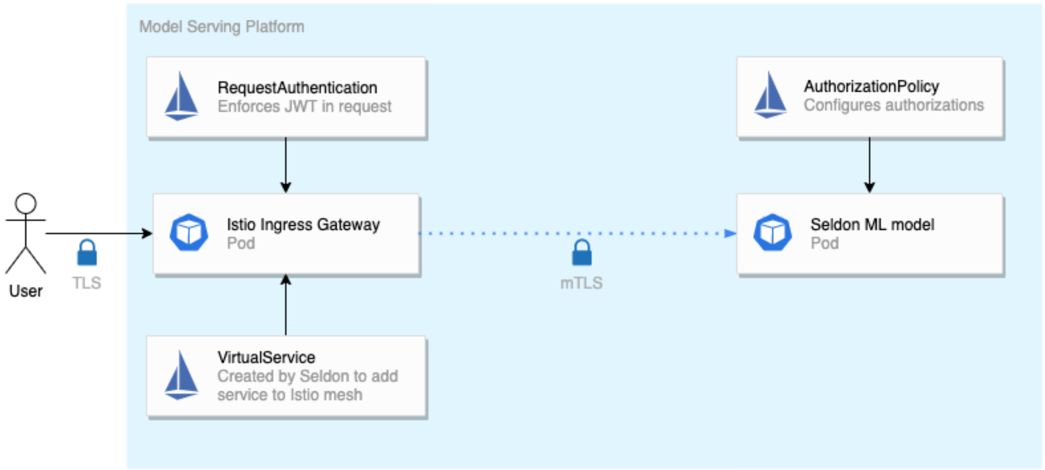
The API is quite simple and supports many of the popular machine learning frameworks like scikit-learn and TensorFlow:

```
from sklearn.ensemble import RandomForestClassifier
from sklearn import datasets
import bentoml
from bentoml.adapters import DataframeInput
from bentoml.frameworks.sklearn import SklearnModelArtifact


def train():
```

Seldon

Seldon Core is a project that allows you to easily deploy machine learning models to a Kubernetes cluster.
because of its support for Istio, the service mesh that we are already using to manage authentication, aut
encryption and monitoring on our platform.



BENTOML

PyTorch
TensorFlow
scikit learn
Keras
dmlc XGBoost
ONNX
fastText
fast.ai
spaCy
Microsoft LightGBM
H2O.ai
Core ML

kubernetes
docker
Amazon SageMaker
Azure Functions
Amazon Lambda
Kubeflow
{REST-API}
NVIDIA
APACHE Spark
KNative
Apache Airflow
Clipper
Prometheus
HEROKU
Grafana
SWAGGER
CloudRun

References