SageMaker Python SDK vs Boto3 SDK

Which SDK To Use And Where With Amazon SageMaker

Ram Vegiraju

Published in

Towards Data Science

.

4 min read

.

Mar 31, 2022

90

1

Image from Unsplash by Hitesh Choudhary

The SageMaker Python SDK and Boto3 Python SDK often lead to a lot of confusion with Amazon SageMaker. At least they did at first with me, until I got some more hands on experience with the service.

Let's quickly establish a clear distinction between the two and then dive into some examples. The SageMaker Python SDK is an open-source library that was developed specifically for training and deploying models on SageMaker. The package is meant to simplify different ML processes on Amazon SageMaker.

The **Boto3 SDK** on the other hand is the **general Python SDK** for AWS. You can use this SDK to interact with any AWS service outside of SageMaker as well. If you prefer languages outside of Python there are a number of corresponding SDKs with the same API calls in popular languages such as Java, Go, and more.

When To Use What

SageMaker SDK

Both these SDKs can be utilized for the same tasks, but in some cases it is more intuitive to use one more than the other. For example SageMaker as a service provides pre-built and maintained images for popular frameworks such as Sklearn, PyTorch, and TensorFlow. You can retrieve these images using the SageMaker SDK and not have to worry about working with Docker. They also help remove any image handling or maintenance by providing ready made estimators for training within these frameworks. Look at this following example to see how easy it is to train using the SageMaker SDK with a supported framework.

Training using the provided SageMaker Estimator for TensorFlow

It’s also super easy to directly deploy from this estimator that you have used for training. The SageMaker SDK takes care of model and endpoint configuration creation by inferring from the parameters you have passed into the estimator. You can directly deploy to an endpoint using a simple “deploy” API call.

Directly Deploy To An Endpoint

In summary it is good practice to use the SageMaker SDK when you have a framework SageMaker supports. Along with this if you are training AND deploying on SageMaker with this framework it is very simple through the use of SageMaker SDK.

Boto3 SDK

On the other hand sometimes you have pre-trained models or different frameworks you may be using. This requires a greater deal of customization and the SageMaker SDK does not always offer that. To demonstrate this let's look at a Bring Your Own Container (BYOC) example with a pre-trained NLP model that you want to deploy for inference. In this case you have an image that you have provided and your own custom framework that does not come out of the box with SageMaker.

We have three important steps and corresponding boto3 API calls that we need to execute to deploy an endpoint: Model Creation, Endpoint Configuration Creation, and Endpoint Creation. The first two entities were abstracted out with the SageMaker SDK with our supported frameworks, but we see those details with the Boto3 SDK.

First we instantiate the clients we will be working with. The sm_client object is what we will be using for the three steps we detailed.
Model Creation
Endpoint Configuration Creation
Endpoint Creation

Using that client we can configure our three API calls with the same details we provided with the SageMaker SDK. The main difference here is that **we have the flexibility to check and customize each of**

the three steps with environment variables or different features. In addition to that we have the power to bring our own framework or pre-trained models or both.

Another major factor to consider is that as you work with more advanced SageMaker offerings such as Multi-Model Endpoints or Multi-Container endpoints it becomes easier to have full-flexibility and control through the Boto3 as you deal with increased customization.

In summary work with the Boto3 SDK as you deal with use-cases that have greater customization. There's a lot more flexibility with this option and as your ML platform becomes more complex you can harness the Boto3 SDK

Conclusion & Additional Resources

I hope this article provided some clarity on the difference of usage between these two SDKs. Once again both can be used for the same purpose or in conjunction a lot of the time, but it can be a lot easier to work with one for specific use-cases. I've attached the entire code for the BYOC and SageMaker SDK examples below.

- BYOC Example Code
- SageMaker SDK TensorFlow Training & Deployment

If you enjoyed this article feel free to connect with me on [LinkedIn](#) and subscribe to my Medium [Newsletter](#). If you're new to Medium, sign up using my [Membership Referral](#).

Sagemaker