SageMaker Model Registry

An Introduction To SageMaker Model Registry

Ram Vegiraju
Published in

Towards Data Science

.
5 min read
.
Mar 25, 2022
68
2

Image from Unsplash by Jeremy Bezanger

It is important to manage different versions of your model through your ML lifecycle. As you train various models you will need to catalog these in a registry of sorts. SageMaker Model Registry helps you manage different model versions and their metadata.

Using Model Registry you can create model package groups that contain different versions of a model. You can create various model package groups for specific sets of models that you may be dealing with. In this article we'll explore an example of training a sample SageMaker XGBoost Regression model and then creating a model package group for the Model Registry. Using this setup we will

walkthrough how you can deploy a model to a SageMaker real-time endpoint directly from the Model Registry.

NOTE: For those of you new to AWS, make sure you make an account at the following **link** if you want to follow along. **This article will assume a novice to intermediate level of knowledge with AWS and SageMaker.**

Table of Contents

1. Setup & Model Training

For this example we will be working in SageMaker Studio, which is the IDE that SageMaker provides. Studio will be our choice over Classic Notebook Instances as it provides UI support for tools such as Model Registry that we will be working with. To get a full walk through on SageMaker Studio setup follow this documentation.

Once in Studio we will be using a base Data Science Kernel with a simple compute instance (ml.c5.large) for our notebook instance. After you've launched your notebook, we will quickly do some setup steps. First we have our AWS SDK Python clients that work with SageMaker.

Using these we will orchestrate the entire process from training to model registry to deployment.

Setup

We can now retrieve the public regression Abalone dataset and upload it to our Studio environment.

Access Abalone Dataset

Now that we have our setup we will proceed to model training. For this article we won't focus on the depths of training, but if you want an article that covers training the XGBoost algorithm on SageMaker in specific check out my other piece here.

We can setup our training input with the Abalone dataset and configure our hyperparameters for the SageMaker XGBoost algorithm we are retrieving.

Training Setup

The key part here is the **image** that we retrieved. This image will be used in our Model Registry to properly catalog the model that we have trained. This metadata with our **model data** and image is crucial for endpoint creation later. Using this we can now train our XGBoost estimator and retrieve the model data from the training job.

Retrieving model data from the training job

2. Model Registry & Deployment

Our first step to working with the Model Registry is creating a Model Group. This Model Group contains a set of versioned models that we will be dealing with for our case. We can create a Model Group using the boto3 client we defined earlier with SageMaker.

Model Group Creation

In this Model Group we have not yet defined any information specific to our model. This is where we pass in the XGBoost image and model data that we are dealing with. This helps provide the necessary metadata for deployment.

Pass in image and model data to the Model Group

In this call we can also specify the type of input data (CSV, JSON) so that our endpoint knows what to expect upon creation. We can see the resource arn and version for this Model Group through the list_model_packages API call.

Model Package ARN

Using the describe_model_package API call we then see all the details we specified.

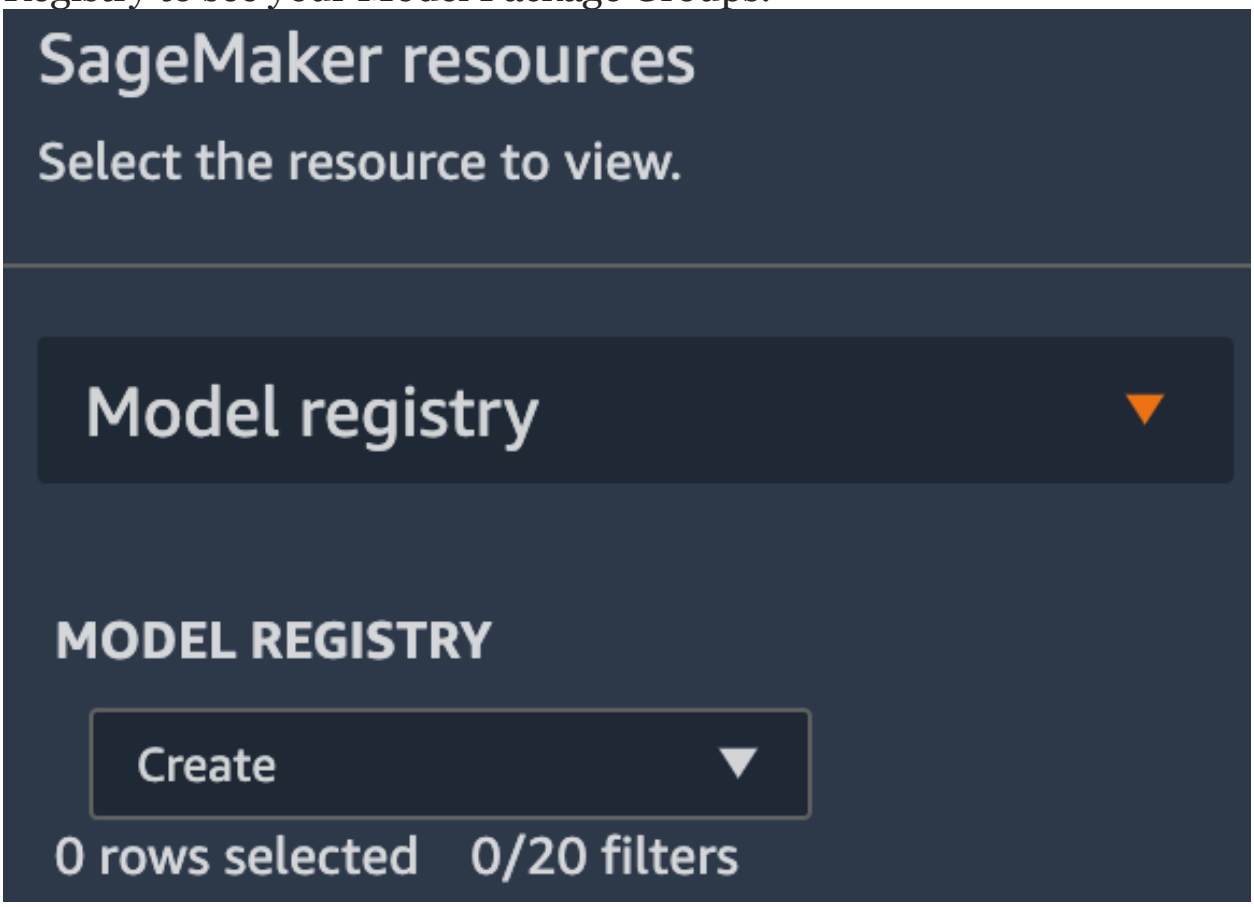
Model Package Details

Now we can approve this model using the model_package arn we retrieved. We can do this by updating the model package status.

Approve model package

The awesome part is you should now also be able to see this model package in the Model Registry tab on Studio. If you select the bottom

most icon on the left panel in Studio you should be able to pick Model Registry to see your Model Package Groups.



Model Registry UI (Screenshot by Author)

If we look closer we can see see our specific group and the version we just approved.

Approved Package (Screenshot by Author)

By approving the model package we are now ready for endpoint creation. This occurs through a series of three steps: SageMaker Model Creation, Endpoint Configuration Creation, and Endpoint Creation. We can work through all these steps by similarly using the boto3 client.

We first create a SageMaker Model entity where we can pass in our model_package arn without having to specify any details about model data or the image used.

SageMaker Model Creation

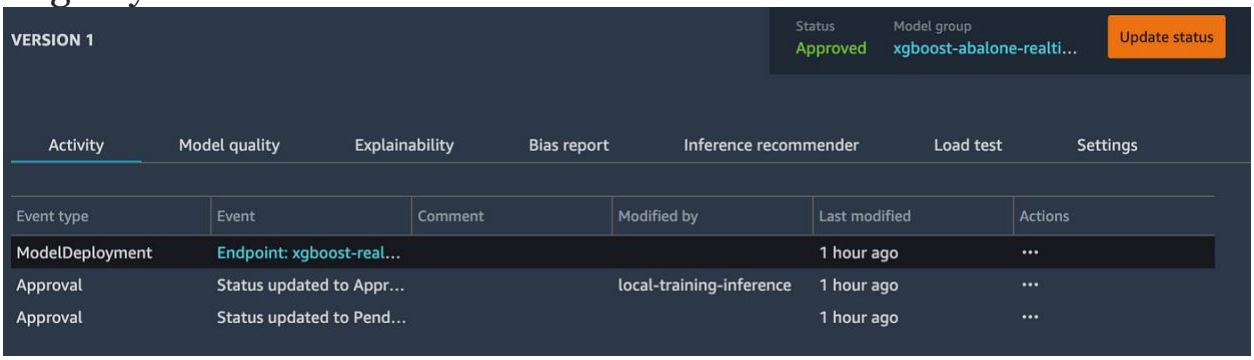
Using this SageMaker Model we can create an Endpoint Configuration which contains details around the instance type and count behind our endpoint.

Endpoint Configuration Creation

Using this endpoint configuration we can feed it into our endpoint and monitor its creation till it is successfully in service.

Endpoint Creation

We can also see this endpoint successfully created in our Model Registry UI.



Endpoint Successfully Created From Model Package (Screenshot by Author)

Once your endpoint has been created we can invoke it with a sample payload and see invocation.

Invocation

```
b'4.566554546356201'
```

Result (Screenshot by Author)

3. Additional Resources & Conclusion

Additional Resources

GitHub - RamVegiraju/SM-Model-Registry: Working with SageMaker Model Registry and the XGBoost…

You can't perform that action at this time. You signed in with another tab or window. You signed out in another tab or…

github.com

For the entire code for the example check out the repository above. I've attached additional examples and resources below.

- Pipelines + Registry Example
- Model Registry Announcement

Conclusion

SageMaker Model Registry is an extremely scalable and powerful tool. We only went through one iteration of training and one model for this

use case. But when you have a model you may be training against multiple datasets, numerous times its true power becomes apparent. It's super easy to group together and catalog different sets of models using the boto3 client to interact with Model Registry.

I hope this article was a good primer with Model Registry, SageMaker, and MLOps. In future articles we will explore how you can incorporate SageMaker Model Registry into an entire MLOps lifecycle with SageMaker Pipelines.

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.