Build a ML Workflow For Scones Unlimited On Amazon SageMaker

REVIEW

CODE REVIEW

HISTORY

Requires Changes

5 specifications require changes

Dear Learner,

Good work, you have followed the correct steps to train, deploy the model and successfully completed the step function workflow

⚠ Following updates required

- Model performance shows the accuracy as less than 50%. However, for this project, it can reach close to 85%. Please check if you have filtered the data correctly.
- Good work on configuring the step function and taking a screenshot, however, please select the output option to check the response from lambda and take the screenshot for submission
- Need to check both positive and negative results in step function

Please check the rubric comments for more details, Looking forward to your next submission

All the best

Further Reading 📚

- Real-time inference
- Bring your own SageMaker image

Train and Deploy a Machine Learning Model

✓

Setup a SageMaker studio and a kernel to run this project

Good work, You have setup the workspace for the project execution.

- Sagemaker provide various kernels and it is important to understand the requirement and use them efficiently with respect to operational requirement and cost

---

Further Reading 📚

- Bring your own SageMaker image
  You can create images and image versions, and attach image versions to your domain, using the SageMaker Studio control panel, the AWS SDK for Python (Boto3), and the AWS Command Line Interface (AWS CLI). You can also create images and image versions using the SageMaker console, even if you haven't onboarded to Studio.

💡 Tips for setting up a SageMaker Studio and a kernel:

- Use a consistent naming convention for your notebooks and code. This will make it easier to find and manage your work.
- Use version control to track your changes. This will help you keep track of your progress and make it easier to revert to previous versions of your work.
- Document your work. This will help you remember what you did and why you did it. It will also be helpful for others who need to understand your work.

✓

Students have completed the ETL (extract, transform, load) section of the starter code.

Well done, you have provided the required steps to perform the ETL process and uploaded the data to S3 bucket

- However, you need to set stream=True for requests.get when downloading the data
- This will help us to download large chunks of data without overlapping and it is best practice

Task
completed the ETL (extract, transform, load) section of the starter code. 🟢

Further Reading 📚

- Define a Pipeline
  To orchestrate your workflows with Amazon SageMaker Model Building Pipelines, you need to generate a directed acyclic graph (DAG) in the form of a JSON pipeline definition

🔁

Students have successfully completed the Model training section up to "Getting ready to deploy", showing they trained an image classification model

Model performance shows the accuracy as less than 50%. However, for this project, it can reach close to 85%. Please check if you have filtered the data correctly and selected the required class as part of model training

Please check and improve the model performance

```
[01/24/2024 19:33:34 INFO 140138517591872] Epoch[26] Time cost=4.862
[01/24/2024 19:33:34 INFO 140138517591872] Epoch[26] Validation-accuracy=0.546875
[01/24/2024 19:33:38 INFO 140138517591872] Epoch[27] Batch [20]#011Speed: 196.549 samples/sec#011accuracy=0.988095
[01/24/2024 19:33:39 INFO 140138517591872] Epoch[27] Train-accuracy=0.985887
[01/24/2024 19:33:39 INFO 140138517591872] Epoch[27] Time cost=4.858
[01/24/2024 19:33:40 INFO 140138517591872] Epoch[27] Validation-accuracy=0.549107
[01/24/2024 19:33:44 INFO 140138517591872] Epoch[28] Batch [20]#011Speed: 193.910 samples/sec#011accuracy=0.985119
[01/24/2024 19:33:45 INFO 140138517591872] Epoch[28] Train-accuracy=0.975806
[01/24/2024 19:33:45 INFO 140138517591872] Epoch[28] Time cost=4.902
[01/24/2024 19:33:46 INFO 140138517591872] Epoch[28] Validation-accuracy=0.541667
[01/24/2024 19:33:49 INFO 140138517591872] Epoch[29] Batch [20]#011Speed: 196.597 samples/sec#011accuracy=0.983631
[01/24/2024 19:33:51 INFO 140138517591872] Epoch[29] Train-accuracy=0.981855
[01/24/2024 19:33:51 INFO 140138517591872] Epoch[29] Time cost=4.872
[01/24/2024 19:33:51 INFO 140138517591872] Epoch[29] Validation-accuracy=0.541667

2024-01-24 19:34:14 Uploading - Uploading generated training model
2024-01-24 19:34:14 Completed - Training job completed
Training seconds: 368
Billable seconds: 368
```

If all goes well, you'll end up with a model topping out above `.8` validation accuracy. With only 1000 training samples in the CIFAR dataset, tl
pursue data augmentation & gathering more samples to help us improve further, but for now let's proceed to deploy our model.

Getting ready to deploy

Further Reading

- Real-time inference
  Real-time inference is ideal for inference workloads where you have real-time, interactive, low latency requirements. You can deploy your model to SageMaker hosting services and get an endpoint that can be used for inference. These endpoints are fully managed and support autoscaling

- Students have successfully completed the "Getting ready to deploy" section, showing they have deployed a trained ML model
- Students have a unique model endpoint name printed in their notebook for use later in the project
- Successfully made predictions using a sample image

This rubric will be evaluated after you have trained the model properly to get better performance

Build a full machine learning workflow

Students have authored three lambda functions.

- 1st lambda is responsible for return an object to step function as `image_data` in an event
- 2nd lambda is responsible for image classification
- 3rd lambada is responsible for filtering low-confidence inferences

Students have saved their code for each lambda function in a python script.

Good work, you have created 3 lambda functions and deployed them. You have shared the lambda function script files as part of the submission

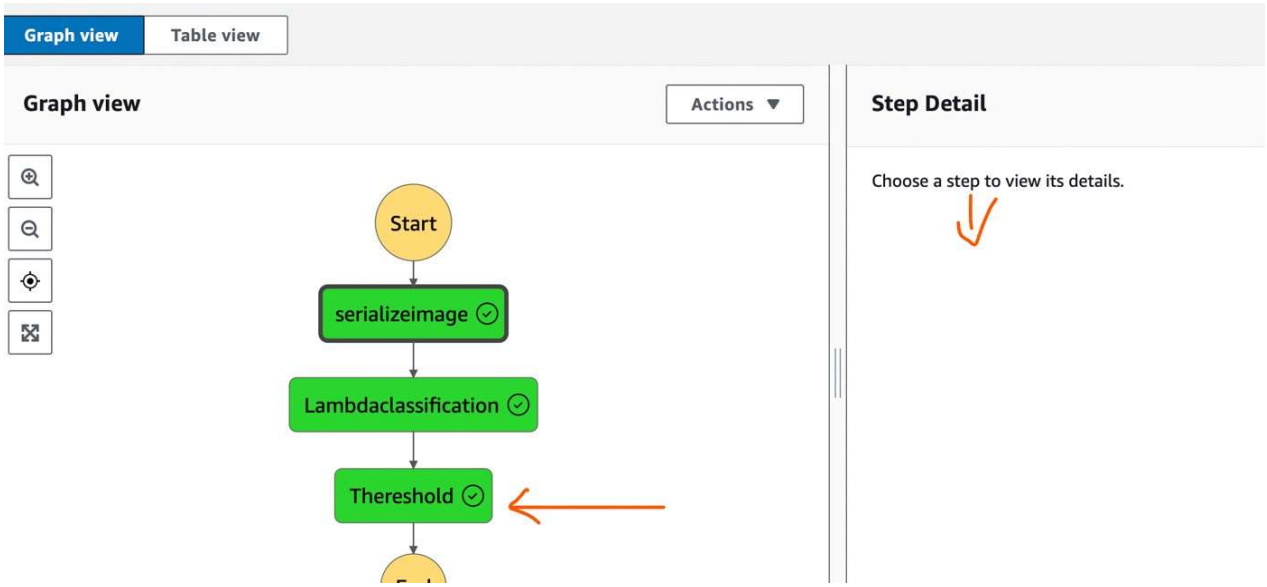
Further Reading

- Best practices for working with AWS Lambda functions
  Use environment variables to pass operational parameters to your function. For example, if you are writing to an Amazon S3 bucket, instead of hard-coding the bucket name you are writing to, configure the bucket name as an environment variable.

- Compose Lambdas together in a Step Function.
- Students will have a JSON export that defines the Step Function
- Students have a screenshot of the working Step Function.

Good work on configuring the step function and taking a screenshot, however, please select the output option to check the response from lambda and take the screenshot for submission

Example Screenshot



Need to check both positive and negative results

You can increase the threshold to see the output when the model output is below the threshold and check the output to see the error message. In this way you can check both success and failure cases

Best practices for composing Lambdas together in a Step Function:

- Use a consistent naming convention for your Lambda functions and Step Functions. This will make it easier to keep track of your code and troubleshoot any problems.
- Use descriptive names for your states. This will make it easier to understand the flow of your Step Function.
- Use the waitForTaskToken property when invoking a Lambda function from a Step Function. This will ensure that the Step Function waits for the Lambda function to finish executing before continuing.
- Use error handling to gracefully handle errors in your Step Function. This will help to prevent your Step Function from failing.
- Use the retry and catch clauses to handle errors in your Lambda functions. This will help to ensure that your Lambda functions are resilient to errors.

- Use the timeout property to set a time limit for your Step Functions and Lambda functions. This will help to prevent your Step Functions and Lambda functions from running for too long

Monitor the model for errors

Students load the data from Model Monitor into their notebook

Correctly done, you have loaded the data from the data capture as part of model monitor into the notebook

This rubric will be evaluated after you have trained the model properly to get better performance

Some best practices for using AWS Model Monitor:

- Choose the right monitoring metrics. Model Monitor provides a variety of metrics to monitor, such as accuracy, precision, recall, and F1 score. Choose the metrics that are most important for your application.
- Configure alerts. Model Monitor can send alerts when your model's performance degrades. Configure alerts so that you are notified as soon as possible of any problems.
- Use explainability features. Model Monitor can provide explanations for your model's predictions. This can help you to understand why your model is making the predictions that it is.
- Monitor your model in production. Once your model is in production, continue to monitor it. This will help you to ensure that your model is performing as expected.

Students create their own visualization of the Model Monitor data outputs

This rubric will be evaluated after you have trained the model properly to get better performance

RESUBMIT

DOWNLOAD PROJECT

Learn the best practices for revising and resubmitting your project.

RETURN TO PATH

Rate this review

START