IS700 FINAL REPORT

- RISHABH SAWA

1. INTRODUCTION:

The goal here is to train a model simple machine learning algorithm in Amazon Sagemaker and deploy the model on an endpoint.We can invoke the model afterwards by calling on this endpoint associated with the model. The data used and model trained data are all stored in the AWS S3 bucket. The REST API is used to invoke a lambda function which calls on the model endpoint to fetch for prediction results for the data queried.
Lets understand the dataset and individual tools used:

A.Dataset:

The dataset used is a simple dataset comprising different species of Iris flower. The dataset consists of 3 classes namely setosa, virginica, versicolor which are actually different species of iris flower(Iris setosa, Iris virginica and Iris versi- color). There are 50 samples for each class. There are four features considered for each sample.

B.Amazon Sagemaker:

AWS Sagemaker is a tool which enables us to build, train and deploy machine learning models with ease. It consists of an integrated jupyter notebook where we can easily manipulate our data and play around it with machine learning algorithms.

C.S3 Bucket:

S3 is an object storage service by AWS. We can store any amount of data for various use-cases. Buckets are more like file folders with data and respective metadata. Every bucket has a global unique name. There are different kinds of permissions and roles which we can have with the help

of AWS Identity and Access Management service commonly known as IAM.

D. Lambda Function:

AWS Lambda function is a computation service which can run remotely any service or application and is also serverless at the sametime. S3 buckets can invoke them for processing the data. We use this function to connect our API and remotely trigger and query results from our model.

E. REST API:

REST API is an API gateway and follows a client server model. Clients can post requests to the server and the server provides a response. We use this REST API to query results from our model.

2. METHODOLOGY:

A: Notebook Instance in Sagemaker:

First, we create a notebook instance in aws sagemaker console. The notebook instance provides us with a jupyter notebook where we prepare our data, train the model and deploy it on an endpoint.

B.Model:

First, we import and prepare our dataset. We are using a dataset which has different species of iris flowers. We have four features and 3 labels. We encode the three classes with an integer value 0,1,2. Following we can see a sample from our dataset with the 4 features and the last column as the species as labeled in the dataset.

The next step is to split the dataset into a training set and test set. In this model, I have split the dataset in the ratio of 7:3 training set and test set.

Then we create an S3 bucket. One thing to note is that each S3 bucket needs to have a unique name globally. Here, I have created a default S3 bucket. We store our training set in the bucket in csv format.

Now, our machine learning model is defined and hyperparameters are set. The model used here is XGBOOST. It is a very efficient supervised learning algorithm for classification problems and regressions.

The algorithm is then trained using the training set stored in S3 bucket. Then the model is deployed to an endpoint.

Then we use the test set to get the predictions and calculate the accuracy of the results by comparing predicted labels against the true labels in test set.

C. Lambda Function:

Our next step is to create and define our lambda function. It can be easily done using aws lambda console. After creating the lambda function with a basic IAM role for lambda, we define how this is going to be used by writing a simple code which would invoke the endpoint which our model was deployed on, get the result for the incoming query and post the result.

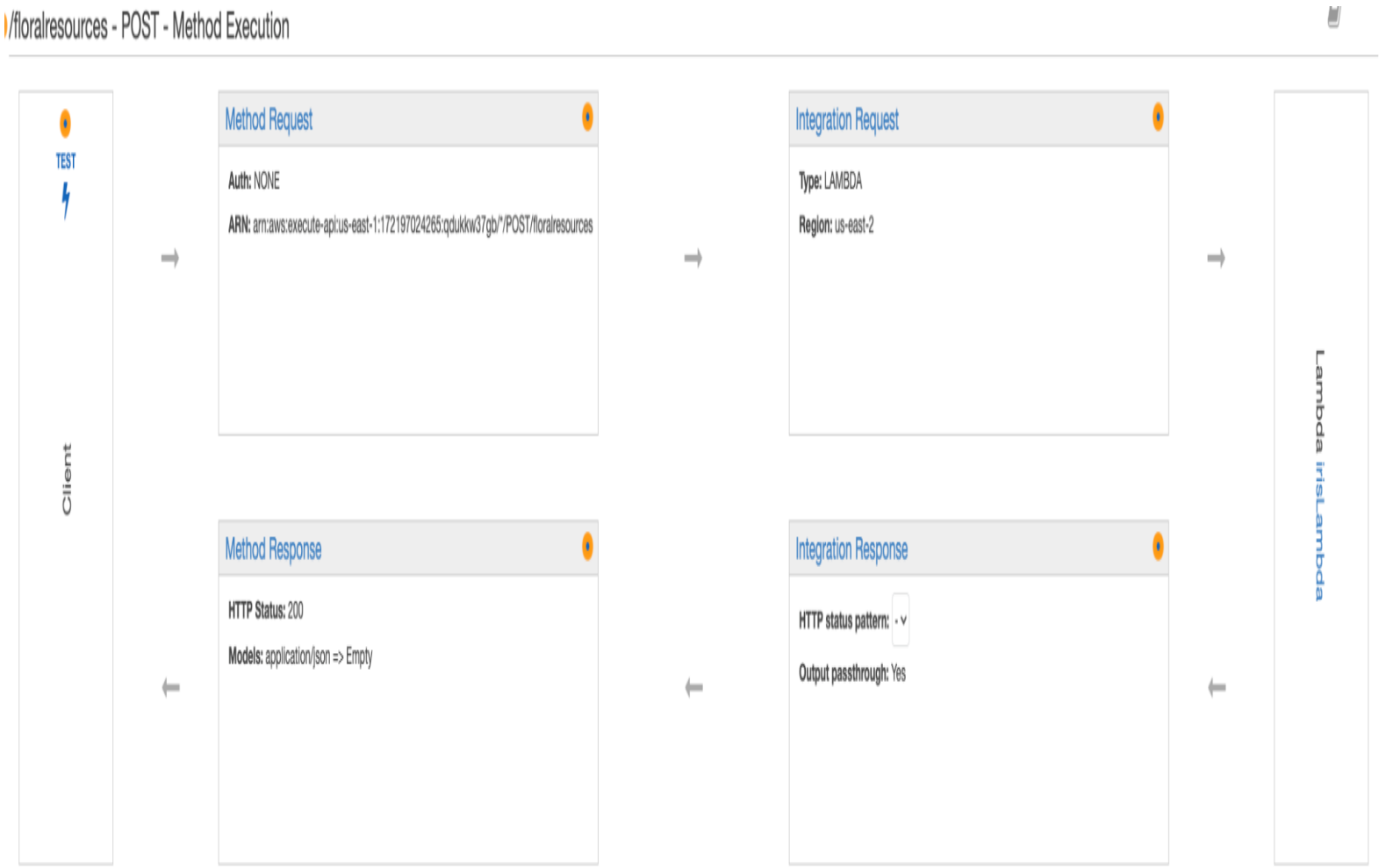
Example for invoking the endpoint :

```
runtime.invoke_endpoint(EndpointName=ENDPOINT_NAME,
                        ContentType='text/csv',
                        Body=payload)
```

Next, we deploy our code, set environment variables. One important thing to note is the roles needed. We need to add a policy for invoking the lambda function in the IAM role. Else, the REST API will only yield errors on posting query.

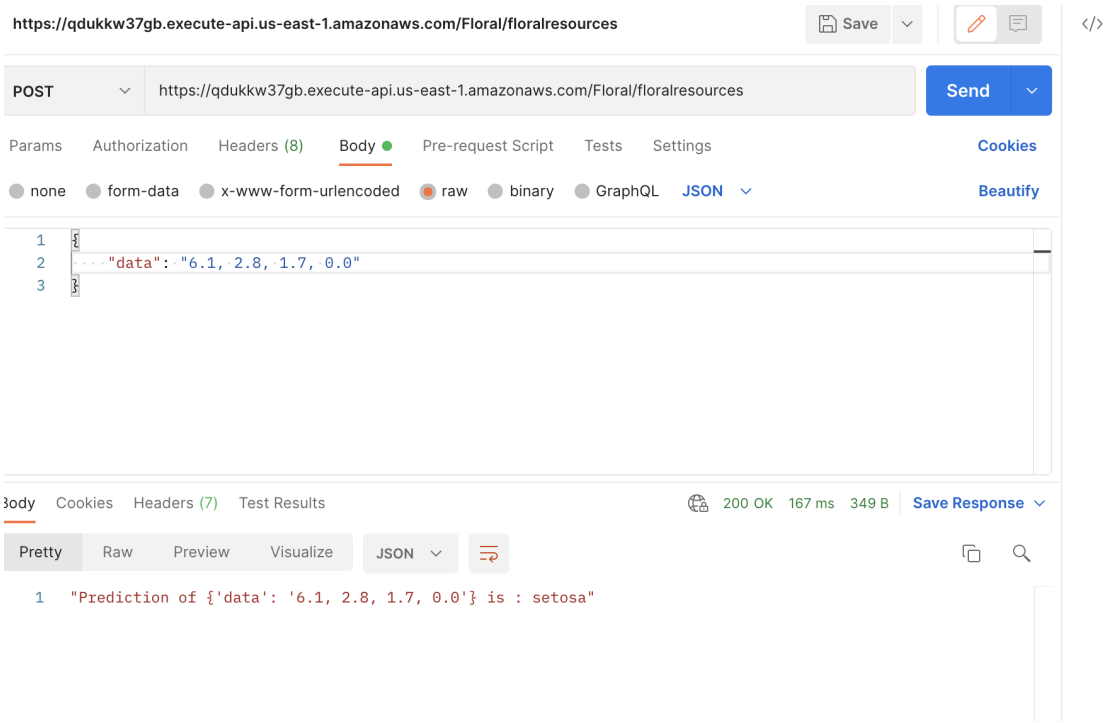
D. REST API:

Next, we need to create a REST API. This is done in the AWS REST API console. The process is quite simple. We can create it in a matter of a few clicks and mention the lambda function earlier in the API. We need to create a Resource and then a method from the action menu. Then from the same actions menu, we deploy the API. After that, we will get an invoke url. This url is used to query and get results.



3. RESULT:

The API can be used to invoke the model and query for results as responses set up in the lambda function. For this, POSTMAN could be used. We use the POST method and pur invoke url. We add a query in json format in the body section and hit send in the postman.



4. Things To Note:

1. This API can work without an api key.
2. It works in the same region.(limitation): This can be defined when creating the lambda function.
3. I created a policy and attached it to the IAM role in the lambda function to allow it to invoke the endpoint. I have mentioned this policy in the Lambda function section in the report.
4. When creating REST API in AWS API Gateway we need to mention the lambda function name and region. When there was a mismatch between the regions of the lambda function API, the REST API was not able to find the Lambda function.(we need to be careful of the region).

5. REFERENCES:

1. https://docs.aws.amazon.com/sagemaker/latest/dg/xgboost.html
2. https://docs.aws.amazon.com/sagemaker/latest/dg/whatis.html
3. https://docs.aws.amazon.com/lambda/latest/dg/lambda-intro-execution-role.html
4. https://aws.amazon.com/blogs/machine-learning/call-an-amazon-sagemaker-model-endpoint-using-amazon-api-gateway-and-aws-lambda/
5. https://docs.aws.amazon.com/sagemaker/latest/dg/xgboost_hyperparameters.html
6. https://www.analyticsvidhya.com/blog/2022/02/building-ml-model-in-aws-sagemaker/
7. https://studymachinelearning.com/train-and-deploy-model-in-aws-sagemaker-create-rest-api-to-call-the-model-using-aws-lambda-and-api-gateway/
8. https://www.kaggle.com/code/khotijahs1/k-means-clustering-of-iris-dataset