Step 1: Create a Dockerfile

First, we need to create a Dockerfile that will set up the necessary environment to run the given Python script.

# Use the official PyTorch image as the base image

FROM pytorch/pytorch:latest

# Set the working directory in the container

WORKDIR /app

# Copy the current directory contents into the container at /app

COPY . /app

# Install any needed packages specified in requirements.txt

RUN pip install --no-cache-dir matplotlib

# Make port 80 available to the world outside this container

EXPOSE 80

# Define environment variable

ENV NAME World

# Run the Python script when the container launches

CMD ["python", "regression.py"]

Step 2: Save the Python Code in a File

Save the Python script of L3 exercise in a file named regression.py.

Step 3: Build the Docker Image

Build the Docker image using the following command:

docker build -t simple-linear-regression .

Step 4: Push the Docker Image to AWS ECR

1. Create a repository in AWS ECR.
2. Tag your Docker image to match the ECR repository.

docker tag simple-linear-regression:latest <your-aws-account id>.dkr.ecr.<region>.amazonaws.com/simple-linear-regression:latest

1. Authenticate Docker to your ECR registry.

aws ecr get-login-password --region <region> | docker login --username AWS --password-stdin <your-aws-account-id>.dkr.ecr.<region>.amazonaws.com

1. Push the Docker image to ECR.

docker push <your-aws-account-id>.dkr.ecr.<region>.amazonaws.com/simple-linear-regression:latest

Step 5: Import the ECR Image into AWS SageMaker

1. Go to the AWS SageMaker console.

2. Navigate to Inference > Models.

3. Create a new model and use the ECR image URI for the container.

Step 6: Deploy the Image with AWS SageMaker

1. Once the model is created, go to Inference > Endpoints.

2. Create a new endpoint configuration and choose the model you just created.

3. Deploy the endpoint.