

Introducing Generative AI with AWS

Project: Building a Domain Expert Model

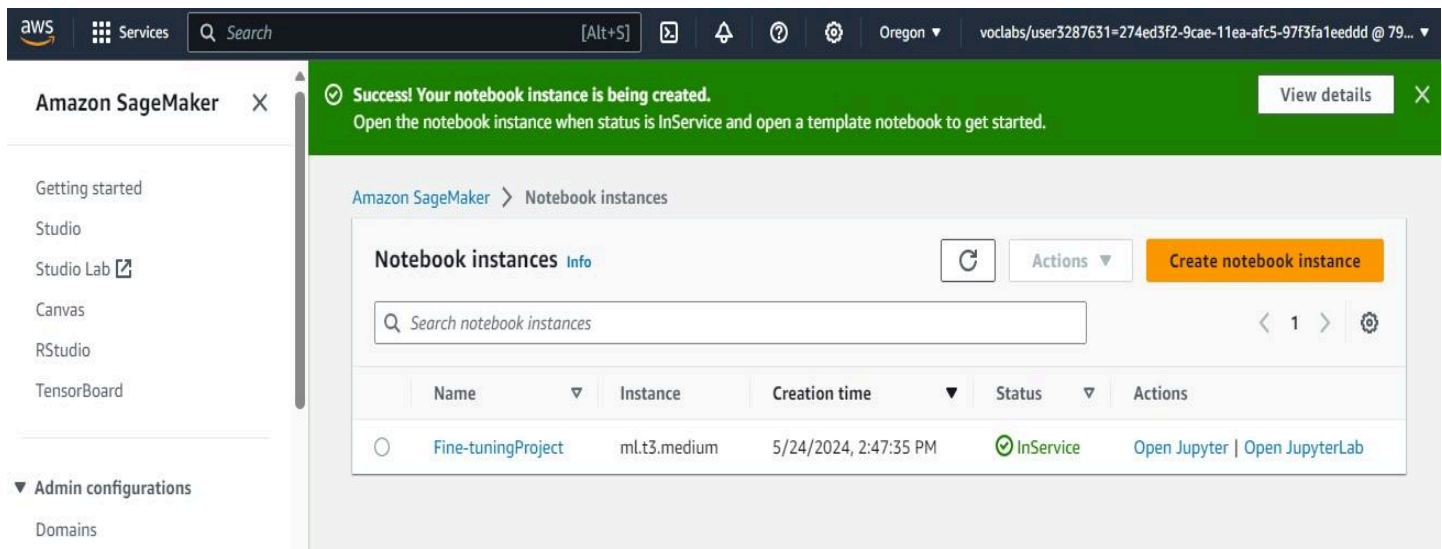
Environment and Project Setup

Configured and completed the below steps and used Aws US West Oregon (us-west-2) Region.

- An AWS SageMaker IAM Role
- An AWS SageMaker Notebook Instance
- A GPU instance for fine-tuning training
- Downloaded the [project starter files](#).

Step 1: Upload Project Starter Files

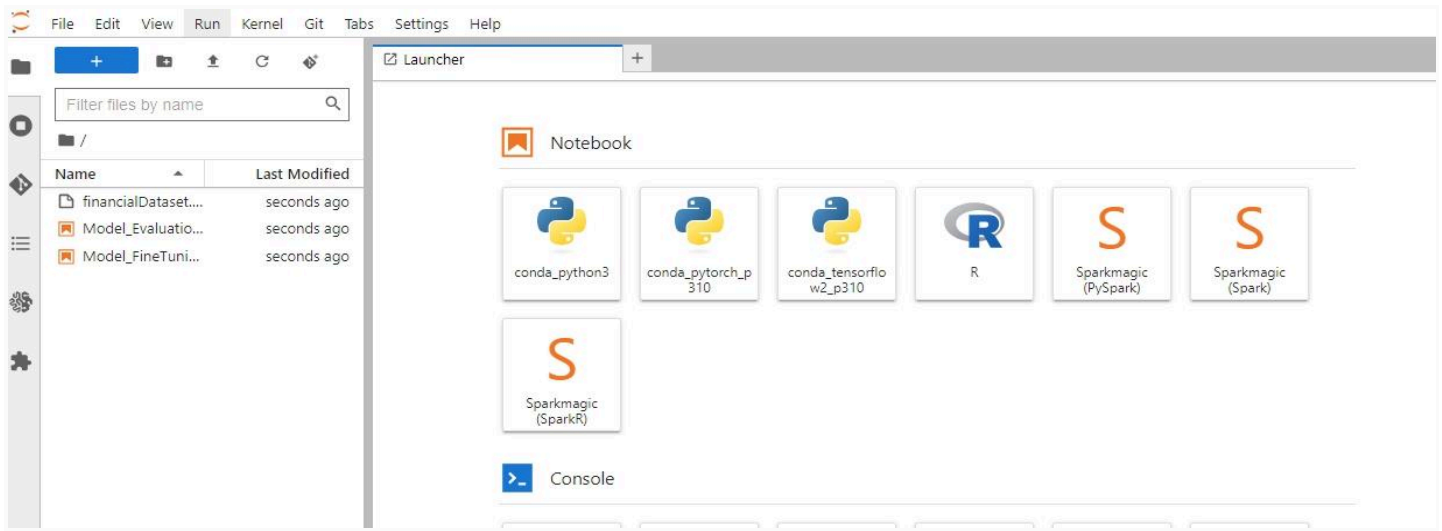
- **Creating and Running the Fine-tuningProject Instance**



The screenshot shows the Amazon SageMaker console interface. At the top, a green banner displays a success message: "Success! Your notebook instance is being created. Open the notebook instance when status is InService and open a template notebook to get started." Below this, the "Notebook instances" section is visible, featuring a search bar and a table of instances. The table contains one instance named "Fine-tuningProject" with the instance type "ml.t3.medium", created on "5/24/2024, 2:47:35 PM", and a status of "InService". The "Actions" column for this instance includes links for "Open Jupyter" and "Open JupyterLab". A sidebar on the left lists navigation options like "Getting started", "Studio", "Studio Lab", "Canvas", "RStudio", "TensorBoard", and "Admin configurations".

Name	Instance	Creation time	Status	Actions
Fine-tuningProject	ml.t3.medium	5/24/2024, 2:47:35 PM	InService	Open Jupyter Open JupyterLab

- **Uploading the Python notebook files (.ipynb)**

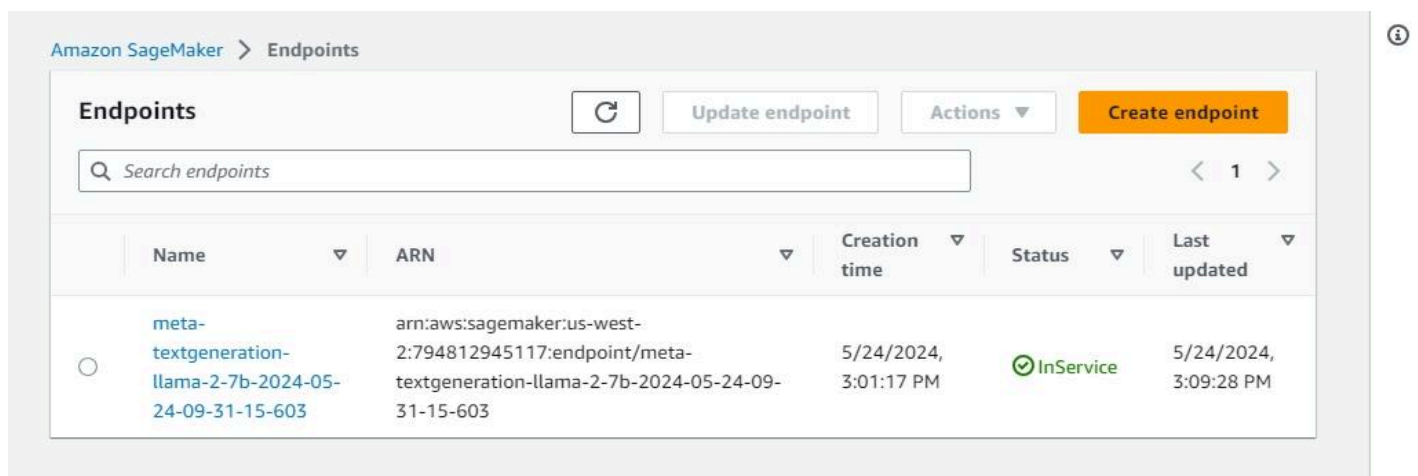


Step 2: Choose your Dataset

I have opted for a dataset within the domain of **Financial Domain**.

Step 3: Deploy and Evaluate the model (Model_Evaluation.ipynb)

- Completed and ran the cells in the Model_Evaluation.ipynb file
- Screenshot of the Model Deployment of Model_Evaluation.ipynb



- Saved and download Model_Evaluation.ipynb with the cell output, uploaded in the zip file and file name is ***Model_Evaluation_UdacityGenAIAWS.ipynb***

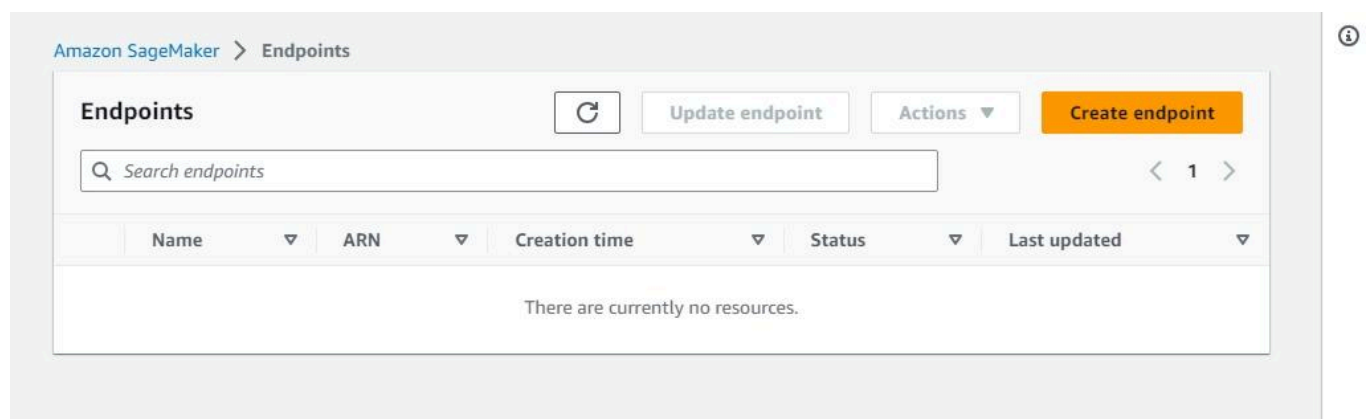
- Screenshot of the Model_Evaluation.ipynb file with the cell output as proof

```
[5]: payload = {
    "inputs": "The results for the short in the money options",
    "parameters": {
        "max_new_tokens": 64,
        "top_p": 0.9,
        "temperature": 0.6,
        "return_full_text": False,
    },
}
try:
    response = predictor.predict(payload, custom_attributes="accept_eula=true")
    print_response(payload, response)
except Exception as e:
    print(e)
```

The results for the short in the money options
 > are also shown in Table 3.
 Table 3. Results for the short in the money options
 The results for the short in the money options are also shown in Table 3.
 Table 3. Results for the short in the money options (continued)
 Table 4. Results for

=====

- Screenshot of Deleted the Model Deployment and endpoints



- Updated the Project Documentation section about the evaluation of the model's text generation capabilities and knowledge.

Step 4: Fine-tune the Model (Model_FineTuning.ipynb)

- Completed and ran the cells in the Model_FineTuning.ipynb

- Saved and download Model_Evaluation.ipynb with the cell output, uploaded in the zip file and file name is ***Model_FineTuning.ipynb***
- screenshot of the Model_FineTuning.ipynb file with the cell output as proof

```
payload = {
  "inputs": "The results for the short in the money options",
  "parameters": {
    "max_new_tokens": 64,
    "top_p": 0.9,
    "temperature": 0.6,
    "return_full_text": False,
  },
}
try:
  response = finetuned_predictor.predict(payload, custom_attributes="accept_eula=true")
  print_response(payload, response)
except Exception as e:
  print(e)
```

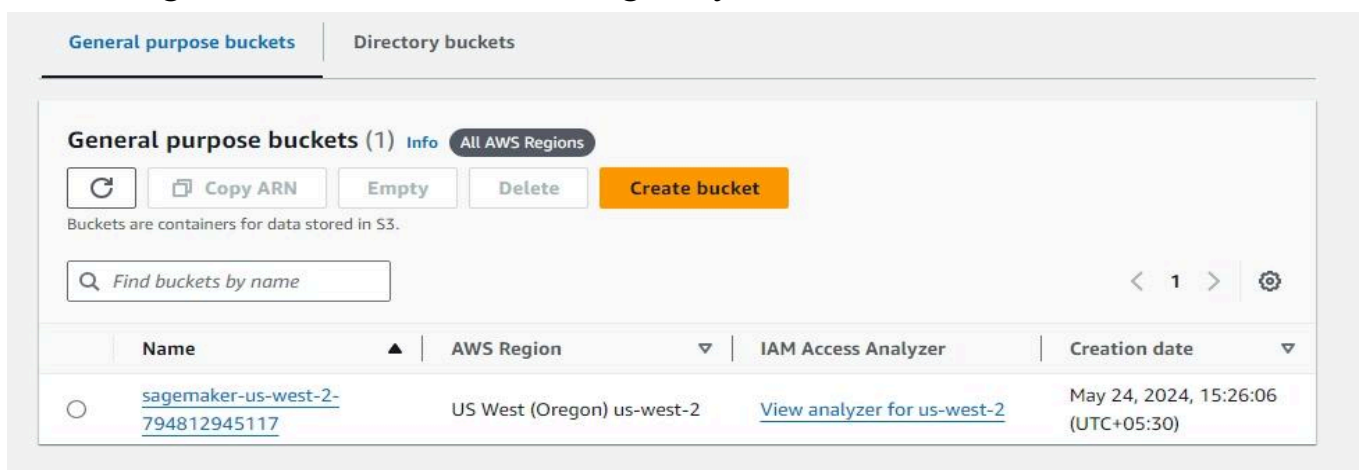
The results for the short in the money options
 > [{"generated_text": ' are shown in Table 16.2.\nTable 16.2. Short In the Money Option Results\nPurchase Price (1.40)\nSale Price (1.40)\nGain (Loss) (\$0.80)\nThe results for the long'}]

=====

- Updated the Project Documentation Report section about fine-tuning the model

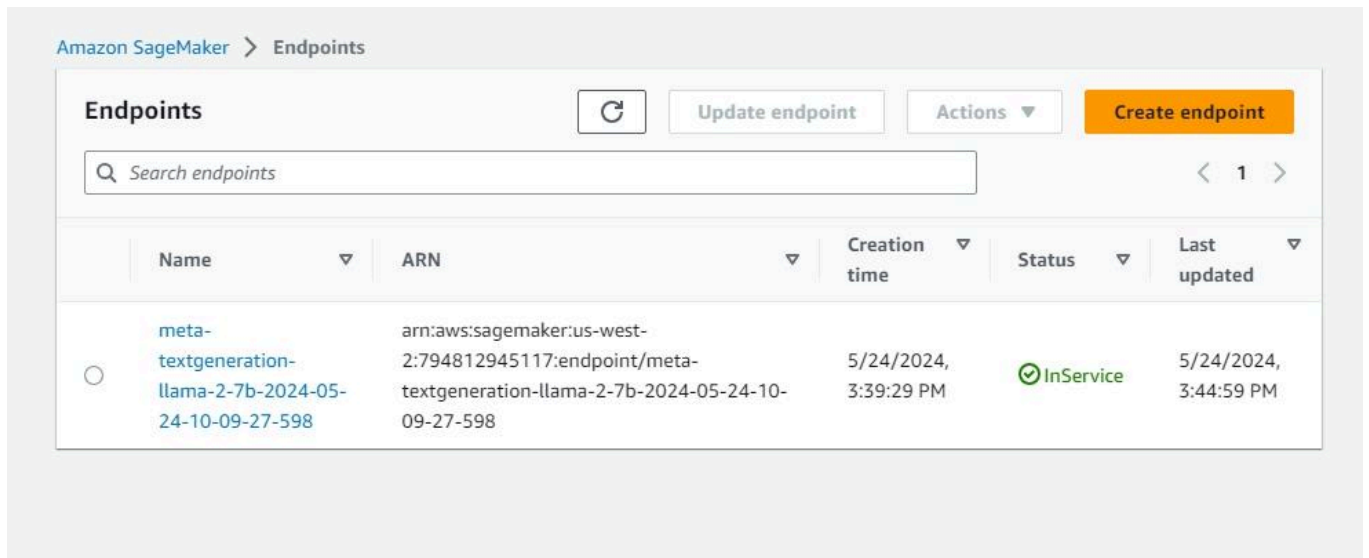
Step 5: Deploy and Evaluate the Fine-tuned Model

- Screenshot of Visiting the AWS S3 bucket where my fine-tuned model weights are stored after training for your submission.



ARN - arn:aws:s3:::sagemaker-us-west-2-794812945117

- Screenshot of Model_FineTuning.ipynb file about deploying and evaluating the fine-tuned model.



- Screenshot of the Model_FineTuning.ipynb file with the cell output as proof

```
[5]: payload = {
    "inputs": "The results for the short in the money options",
    "parameters": {
        "max_new_tokens": 64,
        "top_p": 0.9,
        "temperature": 0.6,
        "return_full_text": False,
    },
}
try:
    response = finetuned_predictor.predict(payload, custom_attributes="accept_eula=true")
    print_response(payload, response)
except Exception as e:
    print(e)
```

The results for the short in the money options
> [{"generated_text": ' are shown in Table 16.2.\nTable 16.2. Short In the Money Option Results\nPurchase Price (0.60)\nSale Price (1.40)\nGain (Loss) (\$0.80)\nThe results for the long'}]

=====

- Screenshot of Deleted the Model Deployment and endpoints(Cell Output)

Use the output from this notebook to fill out the "model fine-tuning" section of the project documentation report

After you've filled out the report, run the cells below to delete the model deployment

IF YOU FAIL TO RUN THE CELLS BELOW YOU WILL RUN OUT OF BUDGET TO COMPLETE THE PROJECT

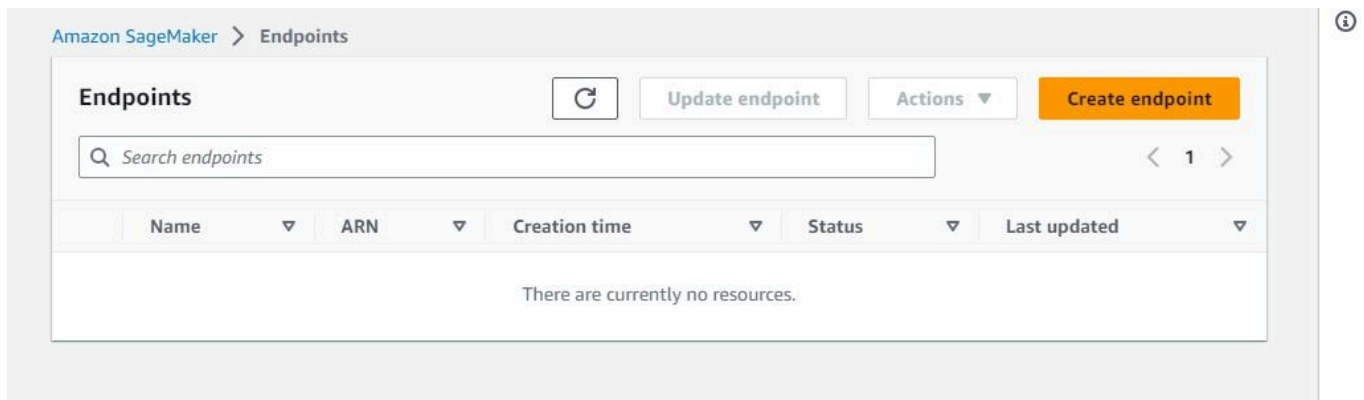
```
[6]: finetuned_predictor.delete_model()
    finetuned_predictor.delete_endpoint()
```

INFO:sagemaker:Deleting model with name: meta-textgeneration-llama-2-7b-2024-05-24-10-09-27-603
INFO:sagemaker:Deleting endpoint configuration with name: meta-textgeneration-llama-2-7b-2024-05-24-10-09-27-598
INFO:sagemaker:Deleting endpoint with name: meta-textgeneration-llama-2-7b-2024-05-24-10-09-27-598

[]:



- Screenshot of Deleted the Model Deployment and endpoints(Cell Output)



- Updated the Project Documentation Report section about fine-tuning the model.

Step 6: Collect Project Documentation and Submit

Zip File Named Project_Building_a_Domain_Expert_Model.zip is uploaded and consists of

- Model_evaluation.ipynb with cell output (File_Name:-Model_evaluation.ipynb).
- Model_FineTuning.ipynb with cell output (File_Name:-Model_FineTuning.ipynb).
- Screenshots of both notebooks with cell output.
- The completed Project Documentation Report (File_Name:- UDACITY Introduction to Generative AI with AWS Project Documentation Report).
- Snapshots folder consists of both the notebooks with outputs for better visibility.