Task 1

 Title

knowledge Wits and Wisdom: Laugh and Learn

 Content

Background

In this task, the developer's initial goal is to familiarize themselves with the "Huggingface LLM Falcon-7b-bf16 Foundation Model" and the "Huggingface LLM Falcon-7b-bf16 Instruction Model," as well as to understand the techniques for evaluating the model's quality both qualitatively and quantitatively.

Your task

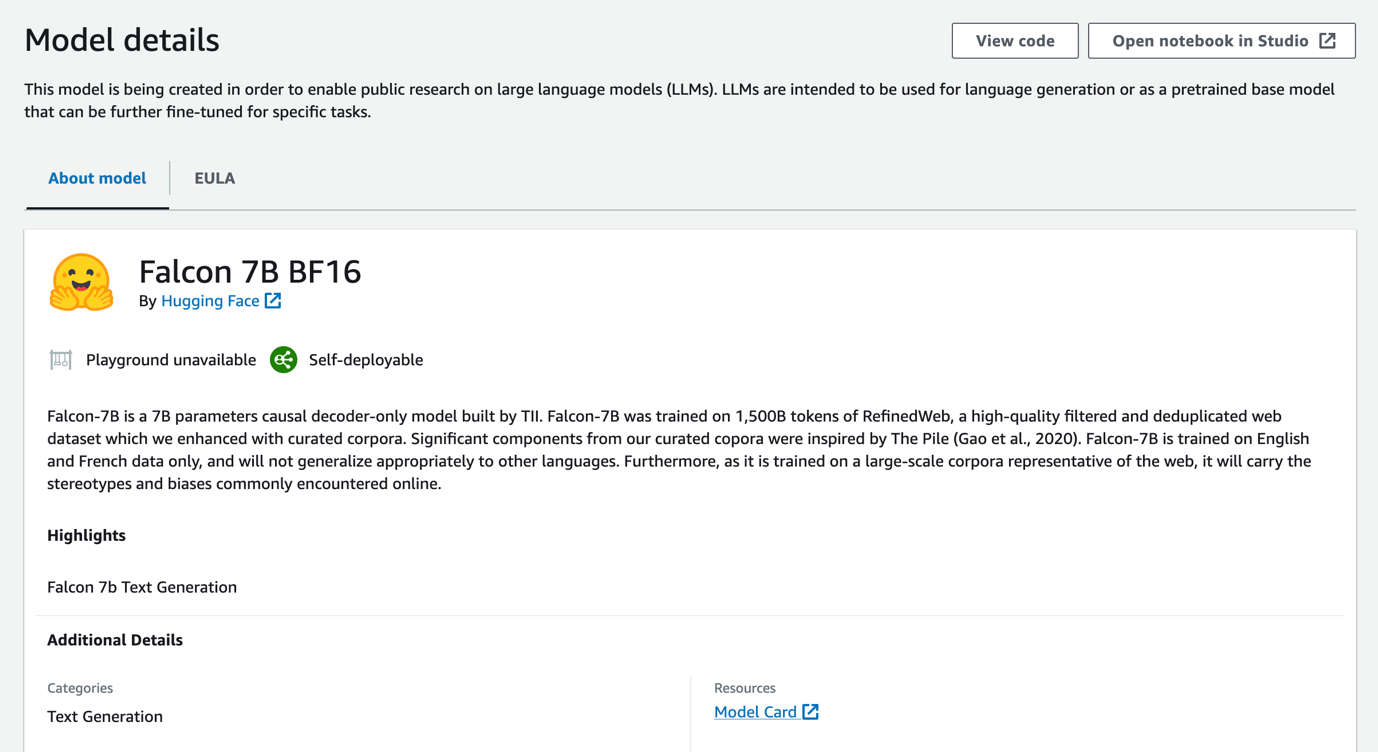
* Detail reading the entire section.

How much time does it take to complete the task

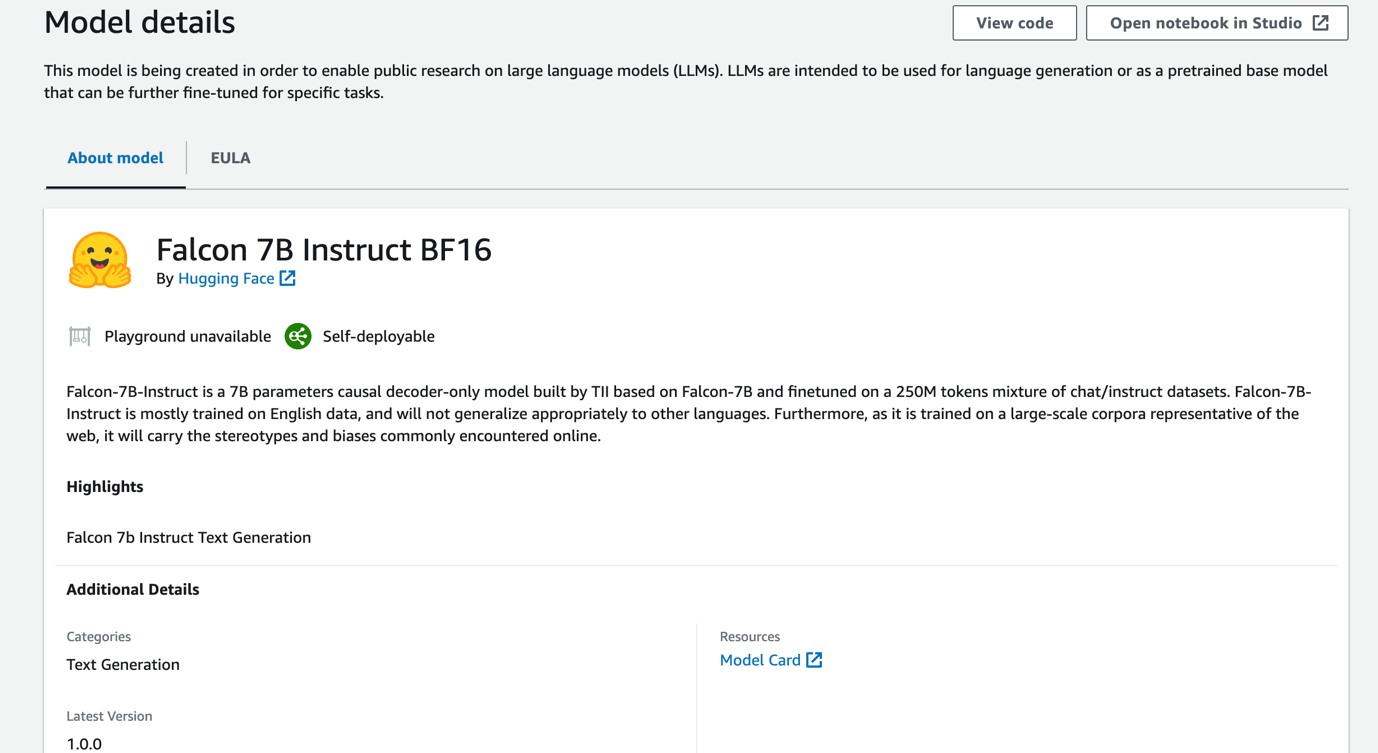
Approximately 30 mins

Getting started

1. Important reading about Huggingface LLM Falcon-7b-bf16 Foundation Model [Falcon-7b-bf16](https://huggingface.co/tiiuae/falcon-7b)



1. Important reading about Huggingface LLM Falcon-7b-instruct-bf16 Foundation Model [Falcon-7b-instruct-bf16](https://huggingface.co/tiiuae/falcon-7b-instruct)



1. Important reading about GenAI LLM model evaluation technique

Some Common Metrics for Evaluating the GenAI LLM Model:

1. **Perplexity**: Perplexity serves as a gauge for assessing the language model's predictive capabilities on a given sample. A lower perplexity score signifies that the model exhibits less confusion when confronted with the test data, indicating superior performance.
2. **BLEU** (Bilingual Evaluation Understudy) Score: The BLEU score is a metric employed to assess the quality of machine-generated text, particularly in contexts such as machine translation. It quantifies the proximity of the model's output to a human reference translation. Higher BLEU scores denote better overall performance.
3. **ROUGE** (Recall-Oriented Understudy for Gisting Evaluation) Score: The ROUGE score comprises a set of metrics utilized in the evaluation of automatic summarization and machine translation. It involves a comparison between the model's output and a set of reference summaries. Elevated ROUGE scores indicate superior performance.
4. **F1 Score**: The F1 score represents the harmonic mean of precision and recall, frequently applied in tasks like named entity recognition and various classification tasks. A higher F1 score signifies better performance.

What you are going to use in this Jam

To assess Large Language Models for Summarization Using ROUGE

Evaluating large language models differs significantly from assessing machine learning models, where metrics such as Accuracy, F1 Score, or Recall were traditionally employed. Metrics for evaluating generated language have their own unique characteristics. Depending on the particular application, diverse metrics are selected to gauge the model's performance. In this jam, you will delve into the application of the ROUGE metric for evaluating the quality of summaries generated by a language model.

ROUGE is not simply a singular metric; it constitutes a collection of metrics designed to quantify the level of overlap and similarity between the generated summary and a reference summary used as a benchmark.

ROUGE yields four distinct individual metrics, namely:

1. **ROUGE-1**: This metric assesses the overlap of unigrams, or single words.
2. **ROUGE-2**: It measures the overlap of bigrams, or pairs of words.
3. **ROUGE-L**: This metric gauges the longest common subsequence, favoring longer shared sequences between the generated and reference summaries.
4. **ROUGE-LSUM**: Computed as the length of the longest common subsequence divided by the sum of the lengths of the generated summary and the reference summary.

Task validation

After completing the reading, write **DONE** to complete this task.

Task 2

 Title

Playing Peekaboo with the Falcon-7b-bf16: Zero Shot and One Shot Shenanigans in Jupyter Notebook with Amazon SageMaker

 Content

Background

In this task, the developer's primary objective is to assess zero-shot and one-shot prompt engineering using the "Huggingface LLM Falcon-7b-bf16 Foundation Model," while also gaining an understanding of the techniques for evaluating the model's quality through both qualitative and quantitative means, specifically using ROUGE metrics.

Objectives of this task

1. Utilizing the Huggingface LLM Falcon-7b-bf16 Foundation Model with Amazon SageMaker through Jupyter Notebook, employing both Zero Shot and One Shot prompting.
2. Assessing Model Evaluation using the "Rouge" metric for the Foundation Model.

How much time does it take to complete the task

Approximately 45 mins

Your Task related with this section

1. Write a python code to find out One shot prompting for dataset index 200 and print the response
2. Write a python code to evaluate rouge metrics for One shot
3. Evaluate differances of rouge1, rouge2, rougeL, rougeLsum for Zero Shot and One shot.

**Hints**: **rouge1(zeroshot)-rouge1(oneshot), rouge2(zeroshot)-rouge2(oneshot), rougeL(zeroshot)-rougeL(oneshot), rougeLsum(zeroshot)-rougeLsum(oneshot)**

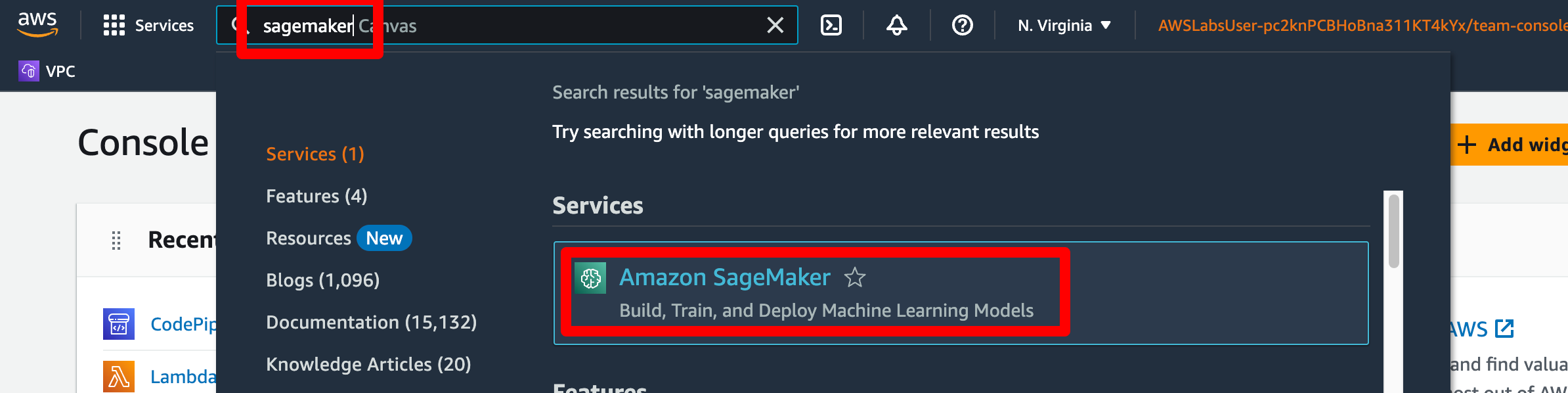
1. Assess whether the above differences are positive or negative.

Getting started

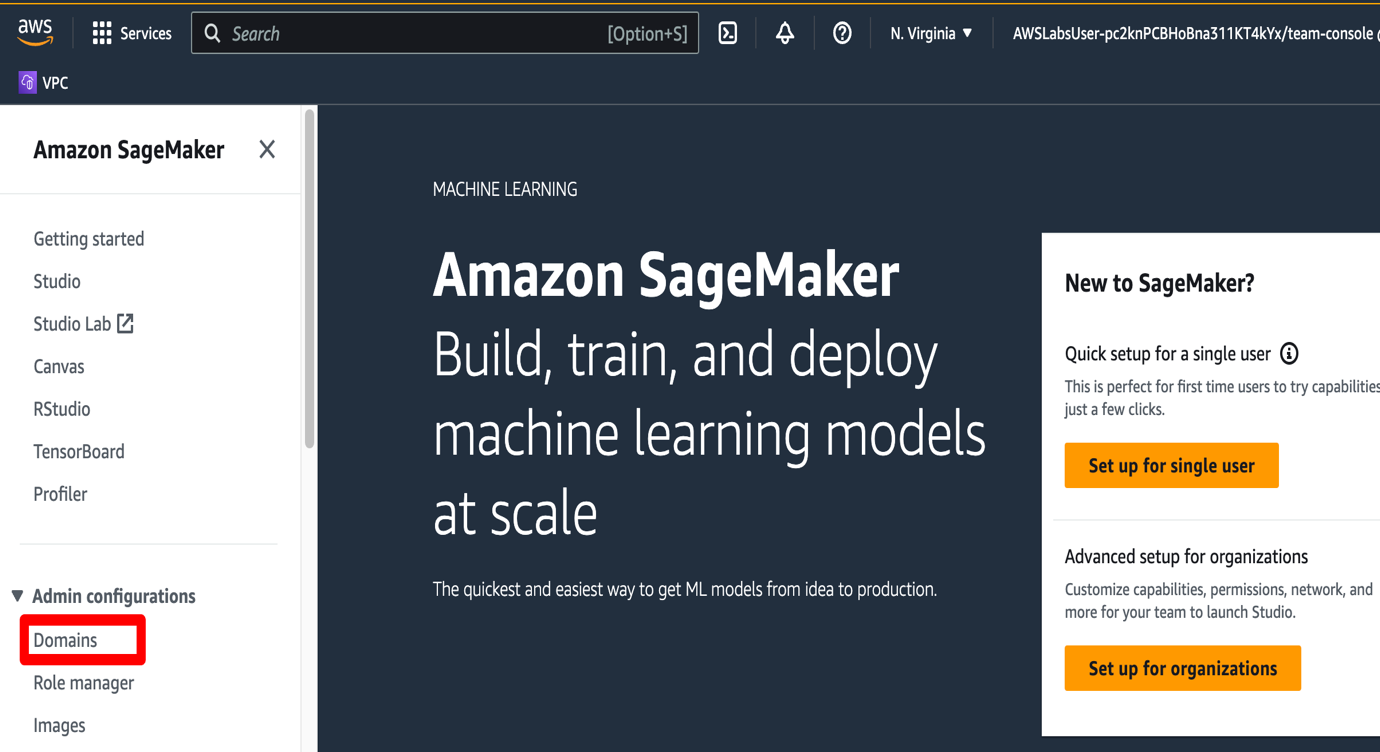
**Request to follow all the below steps [ 1 - 19 ] before starting above subtasks.**

Working Environment set up process:

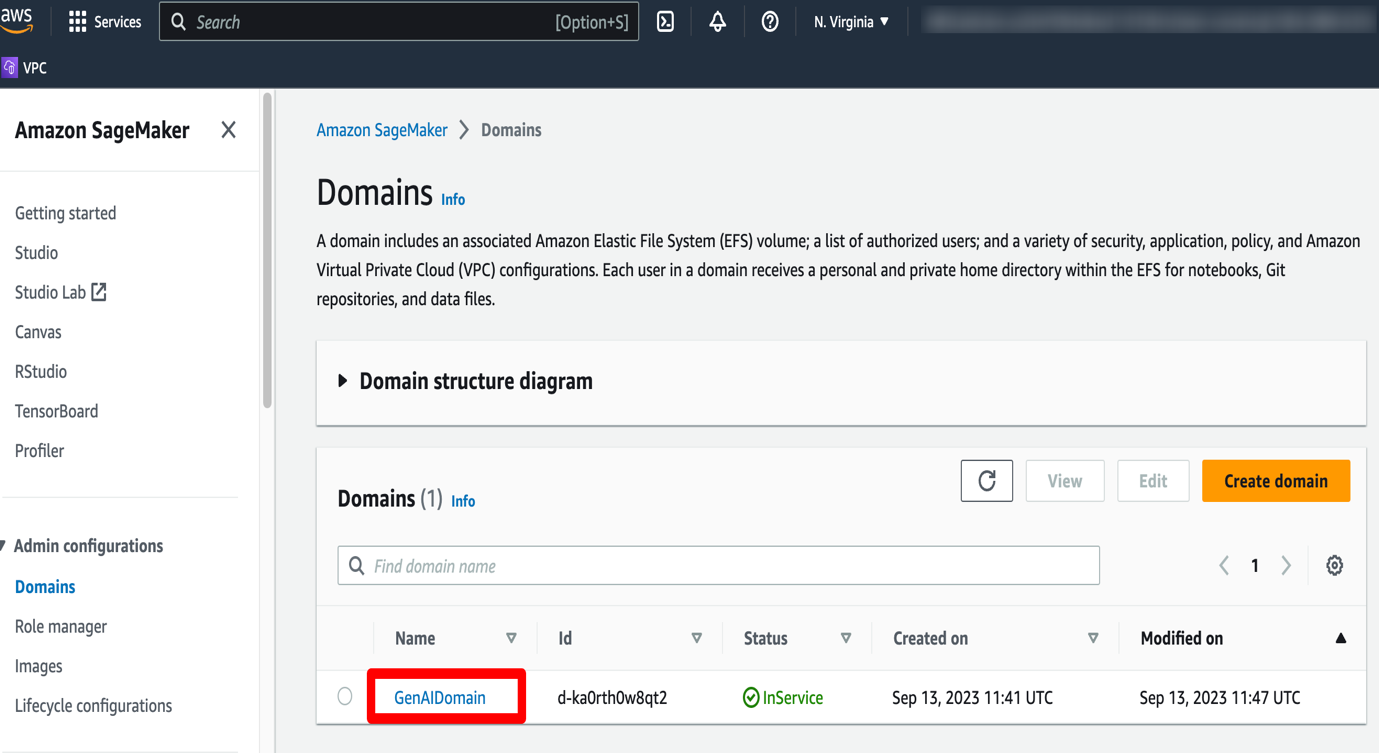
1. In the AWS Management Console search bar, type "*sagemaker*". Then, select **Amazon SageMaker**.



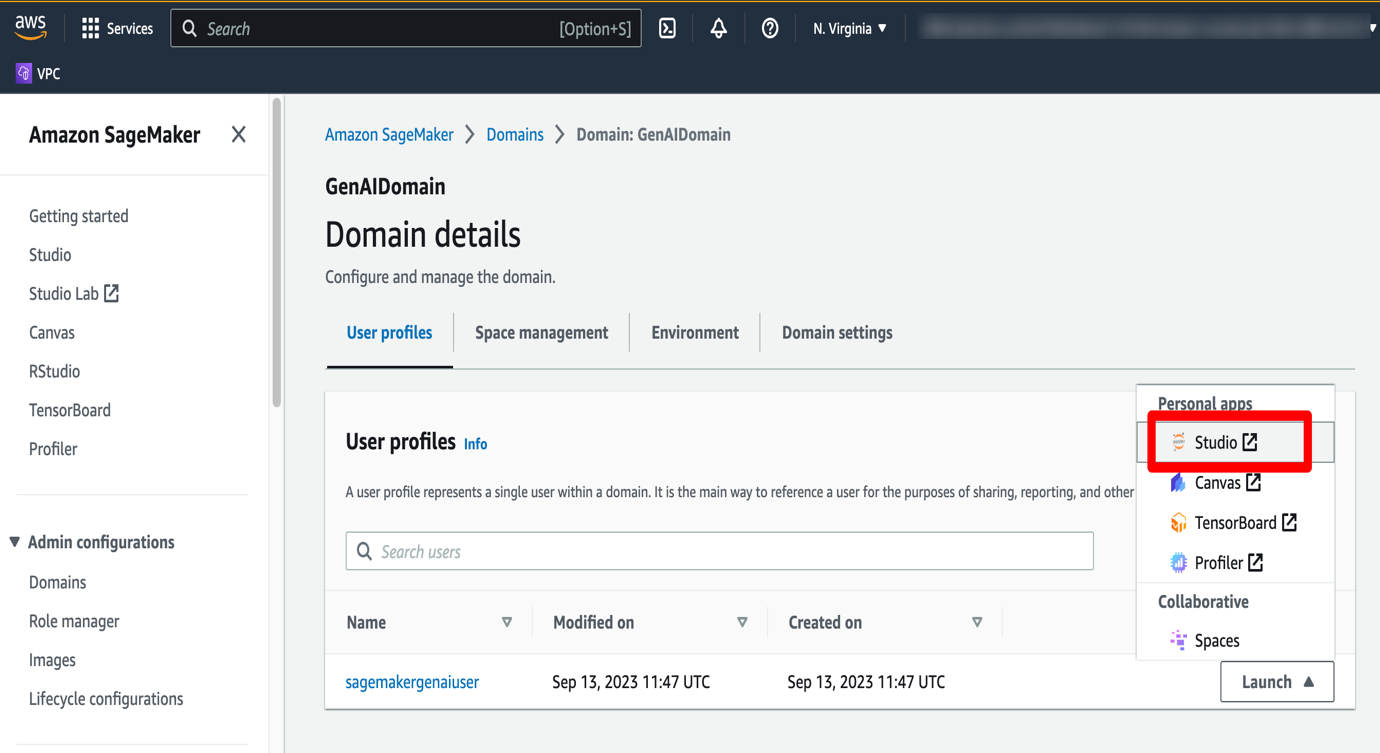
1. In the Amazon SageMaker Console, click **Domains** in the left-side navigation menu.



1. Click **GenAIDomain** in the **Domains** section.



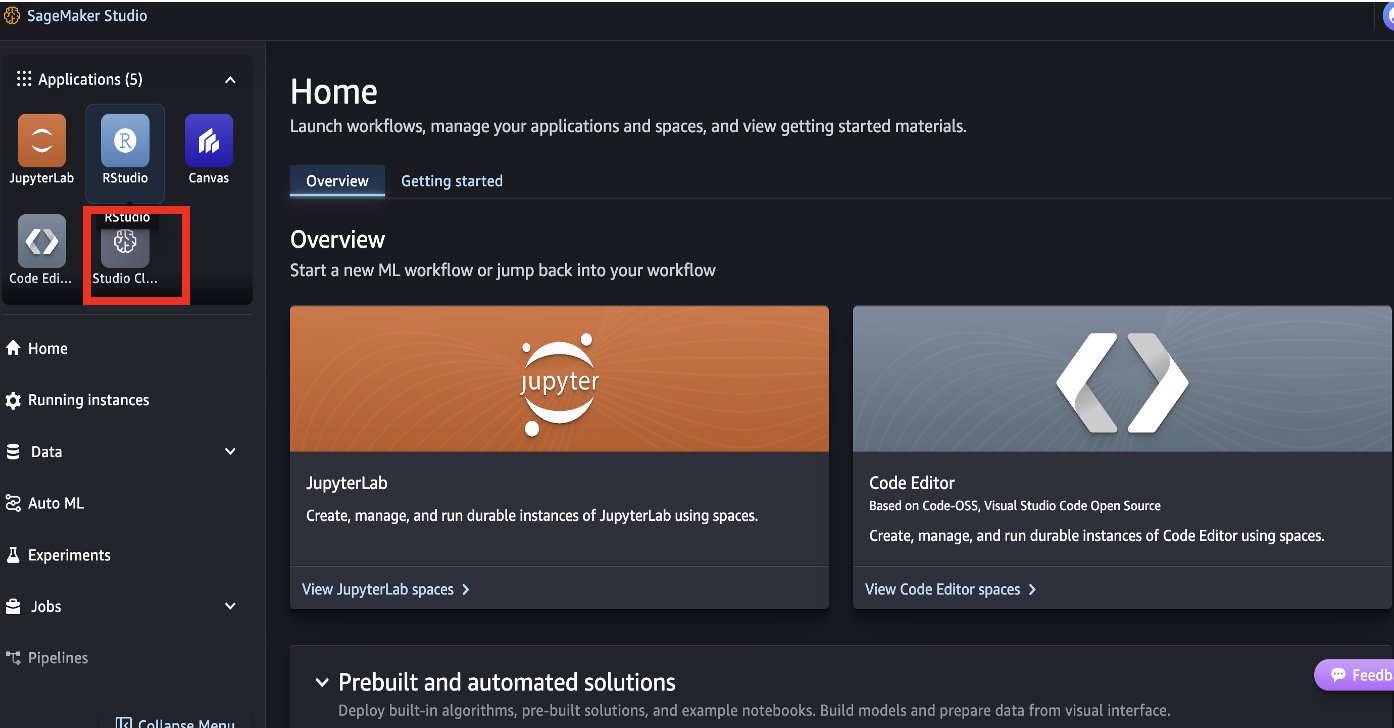
1. Click **Launch**. Click **Studio**.



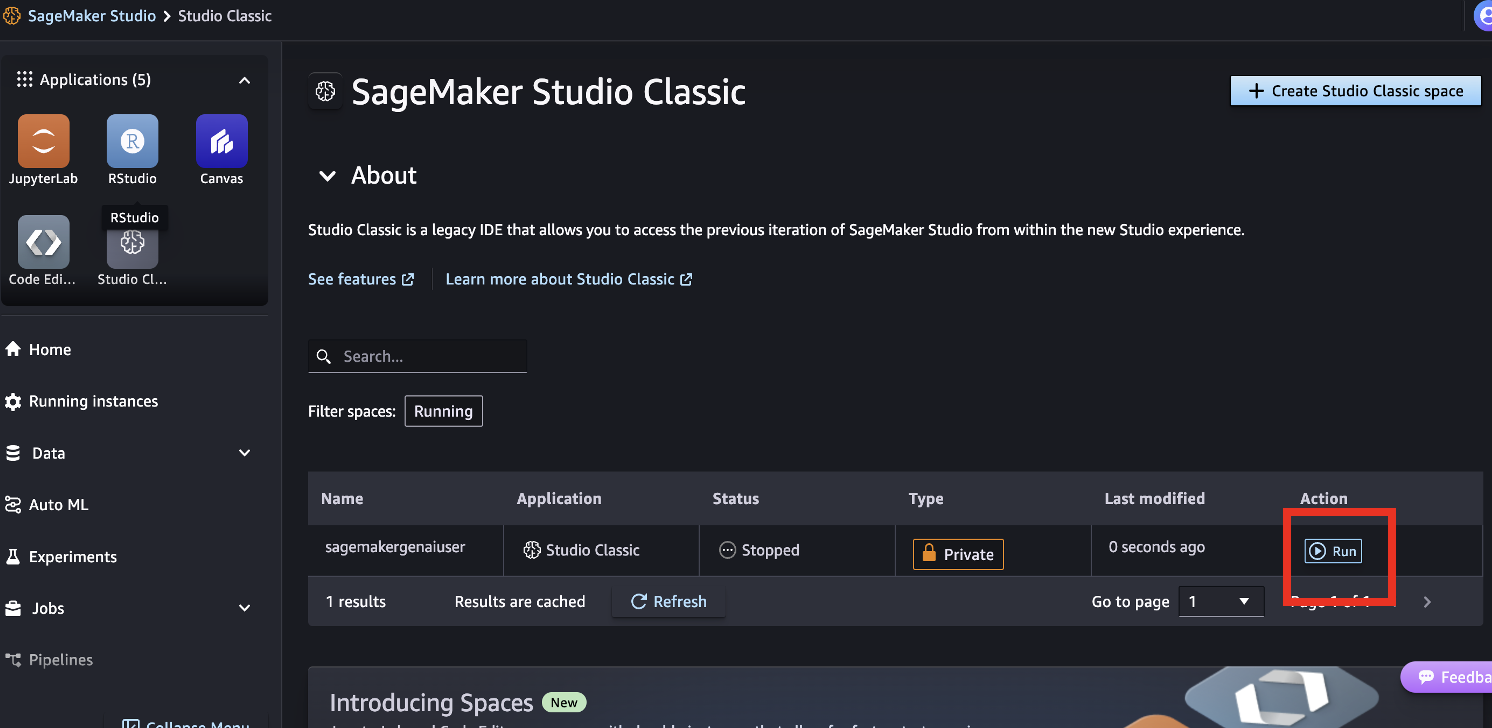
1. Wait for the **Amazon SageMaker Studio** to come online. It will take **~2-3 mins** to open **Amazon SageMaker Studio**.



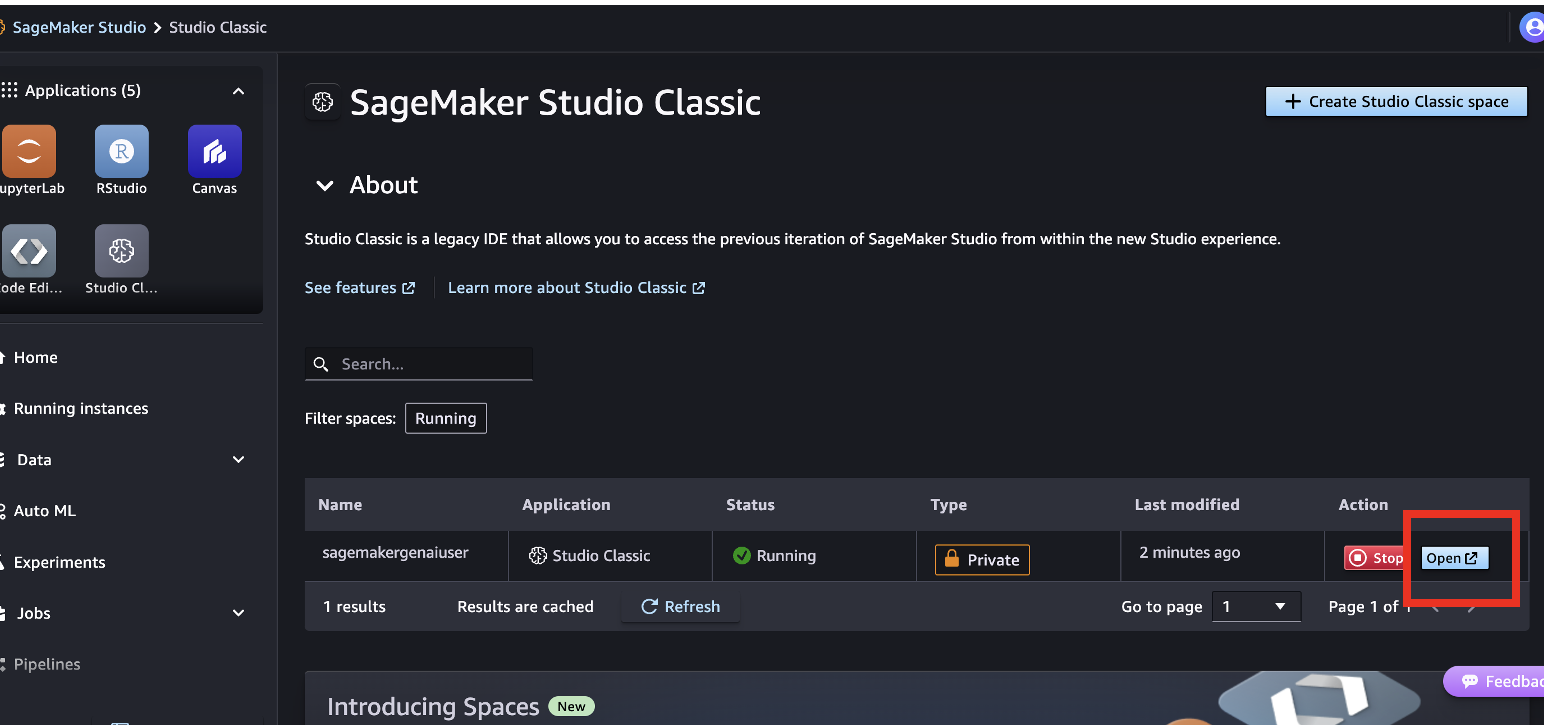
1. This takes you to **Amazon SageMaker Studio**. Click **Studio Clasic**.



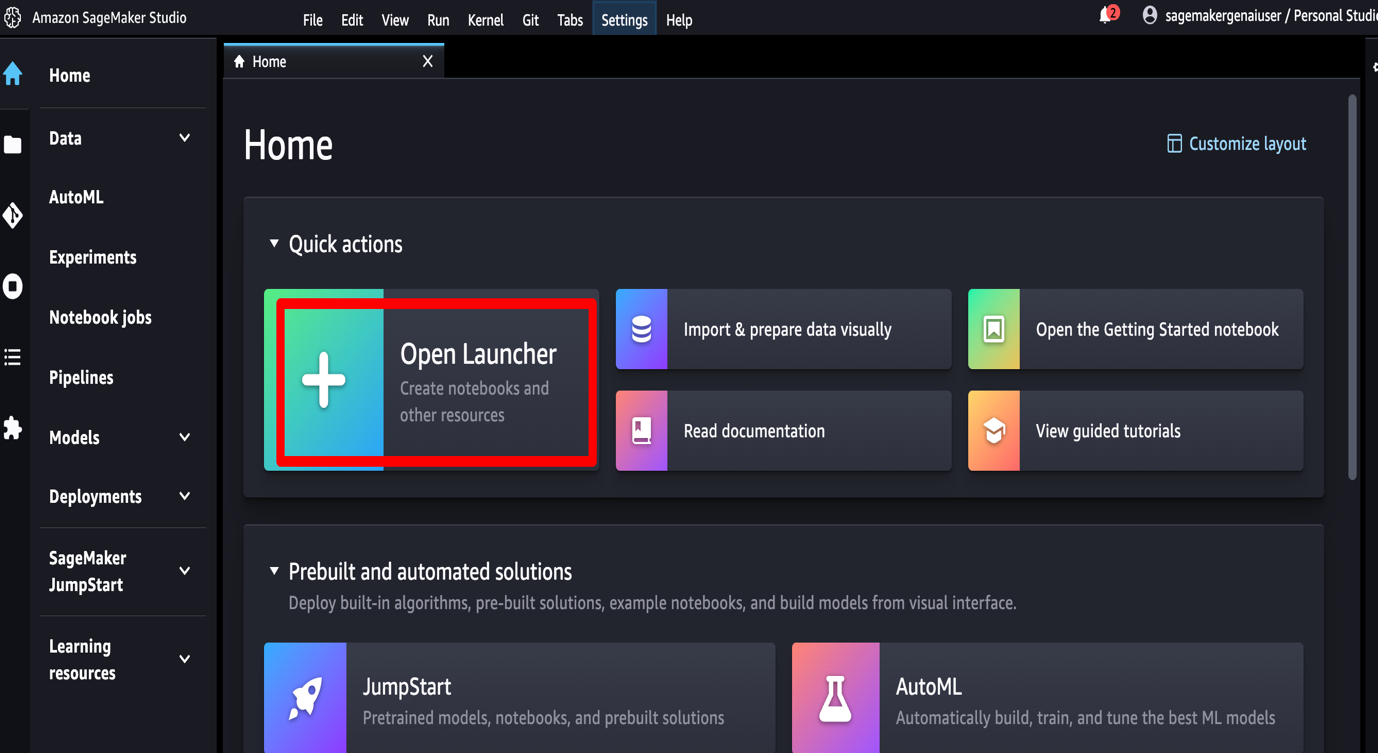
1. Click **Run** in the **Amazon SageMaker Studio**.



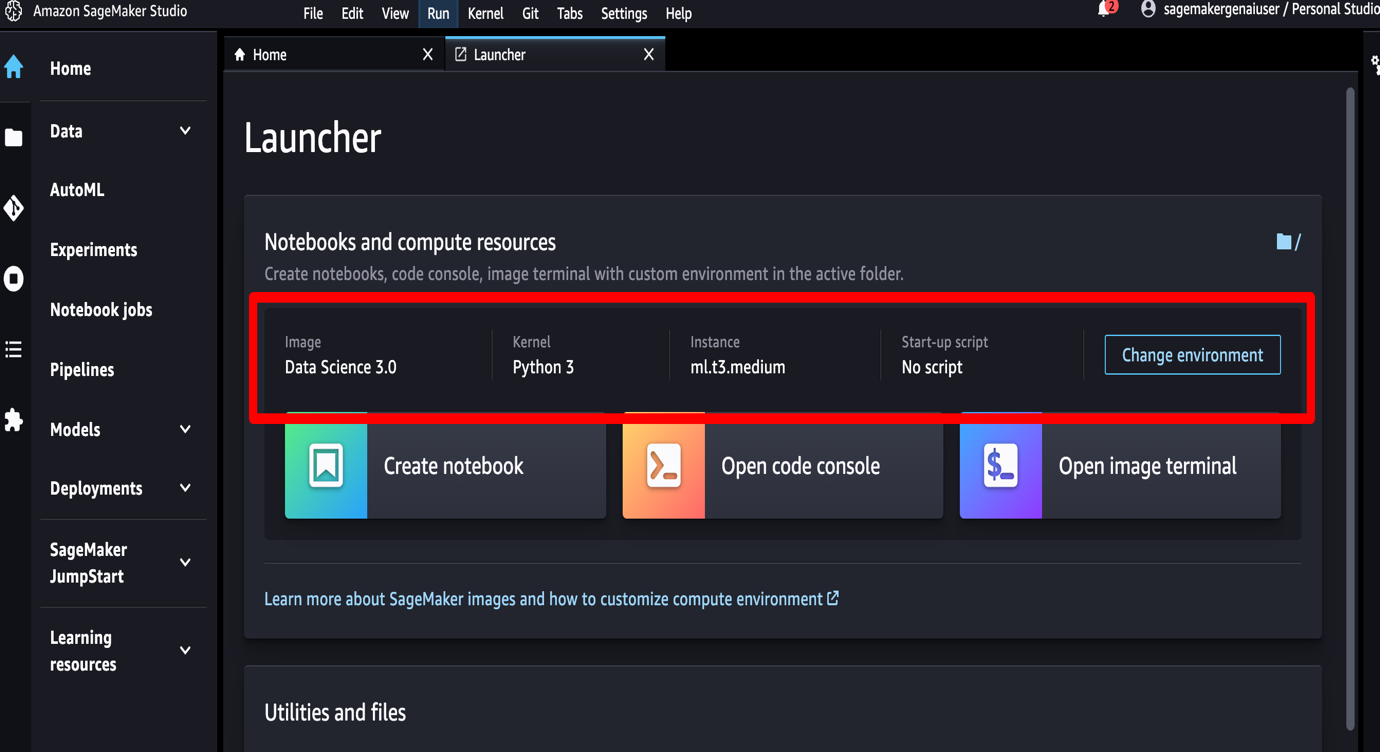
1. Click **Open** in the **Amazon SageMaker Studio**.



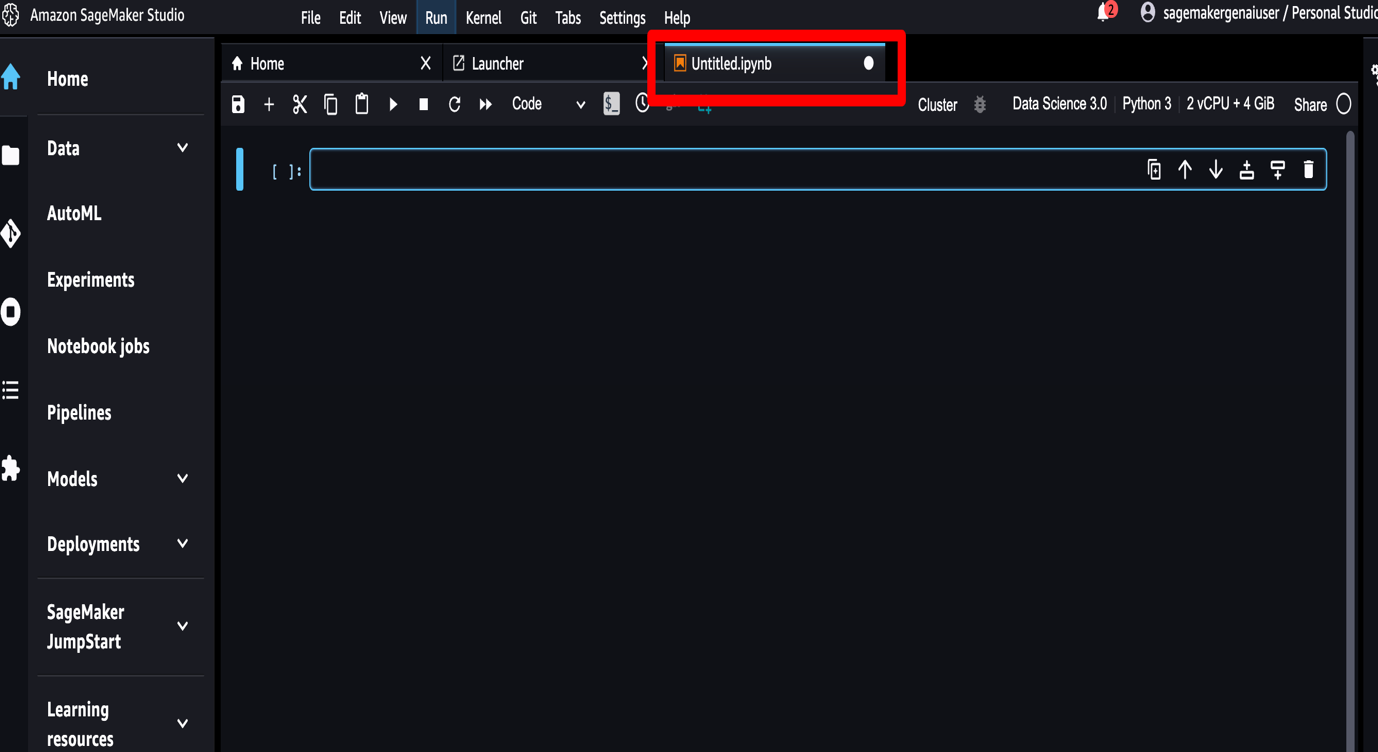
1. This takes you to **Amazon SageMaker Studio**.



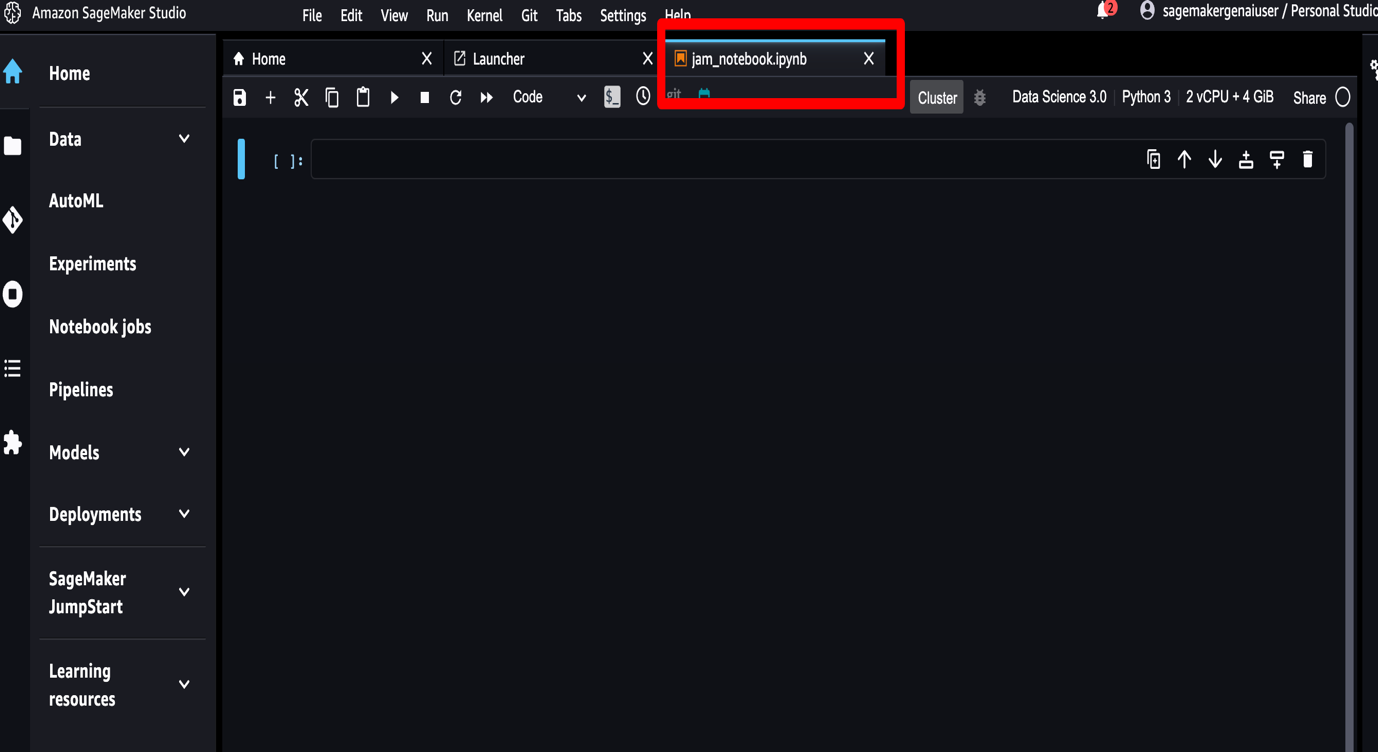
1. Use *Image* as **Data Science 3.0**. *Kernel* as **Python 3**. *Instance* as **ml.t3.large**. Click **Create Notebook**.



1. It opens *Jupyter Notebook*. Wait for the notebook to come online.



1. Rename *Jupyter Notebook* to **jam\_notebook.ipynb**.



1. The *purpose* of the below step to install the required libraries, including the Hugging Face libraries. Copy the below python code. Paste in the first cell of the notebook. Execute the first cell.

Ignore WARNING and note during execution of the below cell. Re execute the cell if any error is coming.

# Install the required libraries, including the Hugging Face libraries, and restart the kernel.

%pip install --upgrade pip

%pip install --disable-pip-version-check \

torch==1.13.1 \

torchdata==0.5.1 --quiet

%pip install \

transformers==4.27.2 \

datasets==2.11.0 \

evaluate==0.4.0 \

rouge\_score==0.1.2 \

loralib==0.1.1 \

peft==0.3.0 --quiet

%pip install -U datasets

1. The *purpose* of the below step to import opensource dataset from huggingface. Copy the below python code. Paste in the second cell of the notebook. Execute the second cell.

# Import opensource dataset from huggingface

from datasets import load\_dataset

huggingface\_dataset\_name = "knkarthick/dialogsum"

dataset = load\_dataset(huggingface\_dataset\_name)

dataset

1. Copy the below python code. Paste in the third cell of the notebook. Execute the third cell. **Note:** Approx execution time **~10 mins**.

%%time

# Import SageMaker JumpStartModel package

from sagemaker.jumpstart.model import JumpStartModel

# Define model\_id and model\_version

model\_id, model\_version, = (

"huggingface-llm-falcon-7b-bf16",

"\*"

)

# deploy the model for prediction

my\_model = JumpStartModel(model\_id=model\_id, instance\_type="ml.g5.4xlarge")

predictor = my\_model.deploy()

1. Copy the below python code. Paste in the fourth cell of the notebook. Execute the fourth cell.

%%time

human\_baseline\_summaries = []

original\_model\_summaries = []

# Zero shot prompting for dataset index 200

index = 200

dialogue = dataset['test'][index]['dialogue']

summary = dataset['test'][index]['summary']

human\_baseline\_summaries.append(dataset['test'][0:10]['summary'])

# Prompt preparation

prompt = f"""

Summarize the following conversation

{dialogue}

Summary:

"""

print(prompt)

payload = {

"inputs": prompt,

"parameters": {

"do\_sample": False,

"top\_p": 0.9,

"temperature": 0.8,

"max\_new\_tokens": 1024,

"stop": ["<|endoftext|>", "<\/s>" ],

},

}

# predict the response

response = predictor.predict(payload)

print(response[0]["generated\_text"])

original\_model\_text\_output = response[0]["generated\_text"]

original\_model\_summaries.append(original\_model\_text\_output)

1. Copy the below python code. Paste in the fifth cell of the notebook. Execute the fifth cell.

# print input prompt, human summary, and model generated - Zero shot

dash\_line = '-'.join('' for x in range(100))

print(dash\_line)

print(f'INPUT PROMPT:\n{prompt}')

print(dash\_line)

print(f'BASELINE HUMAN SUMMARY:\n{summary}\n')

print(dash\_line)

print(f'MODEL GENERATION - ZERO SHOT:\n{original\_model\_text\_output}')

1. Copy the below python code. Paste in the sixth cell of the notebook. Execute the sixth cell.

# print evaluate package to create rouge

import evaluate

rouge = evaluate.load('rouge')

1. Copy the below python code. Paste in the seventh cell of the notebook. Execute the seventh cell.

# Evaluate rouge metrics for Zero shot

original\_model\_results = rouge.compute(

predictions=original\_model\_summaries,

references=human\_baseline\_summaries[0:len(original\_model\_summaries)],

use\_aggregator=True,

use\_stemmer=True,

)

print('ORIGINAL MODEL:')

print(original\_model\_results)

**Please proceed to complete all the tasks outlined in the section titled 'Your Tasks Related to This Section.' Once completed, identify the Amazon SageMaker endpoint and submit this task along with the endpoint name to receive credit.**

1. Use the below one shot prompt for all the tasks you are going to perform.

prompt = f"""

Summarize the following conversation.

#Person1#: Have you considered upgrading your system?

#Person2#: Yes, but I'm not sure what exactly I would need.

#Person1#: You could consider adding a painting program to your software. It would allow you to make up your own flyers and banners for advertising.

#Person2#: That would be a definite bonus.

#Person1#: You might also want to upgrade your hardware because it is pretty outdated now.

#Person2#: How can we do that?

#Person1#: You'd probably need a faster processor, to begin with. And you also need a more powerful hard disc, more memory and a faster modem. Do you have a CD-ROM drive?

#Person2#: No.

#Person1#: Then you might want to add a CD-ROM drive too, because most new software programs are coming out on Cds.

#Person2#: That sounds great. Thanks.

Summary:

#Person1# teaches #Person2# how to upgrade software and hardware in #Person2#'s system.

#############################

Summarize the following conversation.

{dialogue}

Summary: """

Task Validation

* Identify the Amazon SageMaker endpoint and submit this task along with the endpoint name to receive credit.

Hints to identify the Amazon SageMaker endpoint

* Go to **Amazon SageMaker**
* Navigate left side panel. Click **Inference**. Click **Endpoints**.

Python Source Code Chuckles

1. Write a python code to find out One shot prompting for dataset index 200 and print the response

%%time

human\_baseline\_summaries = []

original\_model\_summaries = []

# One shot prompting for dataset index 200

index = 200

dialogue = dataset['test'][index]['dialogue']

summary = dataset['test'][index]['summary']

human\_baseline\_summaries.append(dataset['test'][0:10]['summary'])

# Prompt preparation

prompt = f"""

Summarize the following conversation.

#Person1#: Have you considered upgrading your system?

#Person2#: Yes, but I'm not sure what exactly I would need.

#Person1#: You could consider adding a painting program to your software. It would allow you to make up your own flyers and banners for advertising.

#Person2#: That would be a definite bonus.

#Person1#: You might also want to upgrade your hardware because it is pretty outdated now.

#Person2#: How can we do that?

#Person1#: You'd probably need a faster processor, to begin with. And you also need a more powerful hard disc, more memory and a faster modem. Do you have a CD-ROM drive?

#Person2#: No.

#Person1#: Then you might want to add a CD-ROM drive too, because most new software programs are coming out on Cds.

#Person2#: That sounds great. Thanks.

Summary:

#Person1# teaches #Person2# how to upgrade software and hardware in #Person2#'s system.

#############################

Summarize the following conversation.

{dialogue}

Summary: """

print(prompt)

payload = {

"inputs": prompt,

"parameters": {

"do\_sample": False,

"top\_p": 0.9,

"temperature": 0.8,

"max\_new\_tokens": 1024,

"stop": ["<|endoftext|>", "<\/s>" ],

},

}

# predict the response

response = predictor.predict(payload)

print(response[0]["generated\_text"])

original\_model\_text\_output = response[0]["generated\_text"]

original\_model\_summaries.append(original\_model\_text\_output)

1. Write a python code to print input prompt, human summary, and model generated for One shot

# print input prompt, human summary, and model generated - One shot

dash\_line = '-'.join('' for x in range(100))

print(dash\_line)

print(f'INPUT PROMPT:\n{prompt}')

print(dash\_line)

print(f'BASELINE HUMAN SUMMARY:\n{summary}\n')

print(dash\_line)

print(f'MODEL GENERATION - ONE SHOT:\n{original\_model\_text\_output}')

1. Write a python code to evaluate rouge metrics for One shot

# Evaluate rouge metrics for One shot

original\_model\_results = rouge.compute(

predictions=original\_model\_summaries,

references=human\_baseline\_summaries[0:len(original\_model\_summaries)],

use\_aggregator=True,

use\_stemmer=True,

)

print('ORIGINAL MODEL:')

print(original\_model\_results)

Task 3

 Title

Continue playing Peekaboo with the Falcon-7b-instruct-bf16: Zero Shot and One Shot Shenanigans in Jupyter Notebook with Amazon SageMaker

 Content

Background

In this task, the developer's primary objective is to assess zero-shot and one-shot prompt engineering using the "Huggingface LLM Falcon-7b-bf16 instruction Model," while also gaining an understanding of the techniques for evaluating the model's quality through both qualitative and quantitative means, specifically using ROUGE metrics.

Objectives of this task

* Employing the Huggingface LLM Falcon-7b-bf16 Instruct Model with Amazon SageMaker via Jupyter Notebook, employing Zero Shot and One Shot prompting.
* Evaluating Model Performance using the "Rouge" metric for the Instruct Model.

How much time does it take to complete the task

Approximately 45 mins

Your Task related with this section

1. Write a python code to find out zero shot prompting for dataset index 200 and print the response
2. Write a python code to evaluate rouge metrics for zero shot
3. Write a python code to find out One shot prompting for dataset index 200 and print the response
4. Write a python code to evaluate rouge metrics for One shot
5. Evaluate differances of rouge1, rouge2, rougeL, rougeLsum for Zero Shot and One shot.

**Hints**: **rouge1(zeroshot)-rouge1(oneshot), rouge2(zeroshot)-rouge2(oneshot), rougeL(zeroshot)-rougeL(oneshot), rougeLsum(zeroshot)-rougeLsum(oneshot)**

1. Assess whether the above differences are positive or negative.

**Request to follow all the below steps before starting above subtasks.**

Getting started

Working Environment set up process:

1. Continue with the same *Jyputer notebook* notebook that you are using for task 2.
2. Exceture the below cell before starting task 3

predictor.delete\_model()

predictor.delete\_endpoint()

**Please proceed to complete all the tasks outlined in the section titled 'Your Tasks Related to This Section.' Once completed, identify the Amazon SageMaker endpoint and submit this task along with the endpoint name to receive credit.**

**Please use model id as huggingface-llm-falcon-7b-instruct-bf16 for task 3**

Common Issues

In case the participants encounter the following error: "The account-level service limit 'ml.g5.4xlarge for endpoint usage' is 1 Instances, with current utilization of 1 Instances and a request delta of 1 Instances. Please use AWS Service Quotas to request an increase for this quota.."

kindly make sure the existing model endpoints are deleted.

* Go to **Amazon SageMaker**
* Navigate left side panel. Click **Inference**. Click **Endpoints**.

Task Validation

Identify the value of rouge1, rouge2, rougeL and rougeLsum from **Jupyter notebook** after completing all the tasks mentioned above. Submit the value in the below format with the value you received after execution for rouge1, rouge2, rougeL and rougeLsum.

rouge1 : x.xxxxx , rouge2 : x.xxxxx, rougeL : x.xxxxx, rougeLsum : x.xxxxx

The value x.xxxxx should be replaced with the original value derived from Jupyter notebook

Please strictly follow the above format to receive credit.

Python Source Code Chuckles

1. Copy the below python code. Execute the cell. **Note:** Approx execution time **~10 mins**.

%%time

# Import SageMaker JumpStartModel package

from sagemaker.jumpstart.model import JumpStartModel

# Define model\_id and model\_version

model\_id, model\_version, = (

"huggingface-llm-falcon-7b-instruct-bf16",

"\*"

)

# deploy the model for prediction

my\_model = JumpStartModel(model\_id=model\_id, instance\_type="ml.g5.4xlarge")

predictor = my\_model.deploy()

1. Copy the below python code. Execute the cell.

%%time

human\_baseline\_summaries = []

original\_model\_summaries = []

# Zero shot prompting for dataset index 200

index = 200

dialogue = dataset['test'][index]['dialogue']

summary = dataset['test'][index]['summary']

human\_baseline\_summaries.append(dataset['test'][0:10]['summary'])

# Prompt preparation

prompt = f"""

Summarize the following conversation

{dialogue}

Summary:

"""

print(prompt)

payload = {

"inputs": prompt,

"parameters": {

"do\_sample": True,

"top\_p": 0.9,

"temperature": 0.1,

"max\_new\_tokens": 1024,

"stop": ["<|endoftext|>", "<\/s>"]

}

}

# predict the response

response = predictor.predict(payload)

print(response[0]["generated\_text"])

original\_model\_text\_output = response[0]["generated\_text"]

original\_model\_summaries.append(original\_model\_text\_output)

1. Copy the below python code. Execute the cell.

# print input prompt, human summary, and model generated - Zero shot

dash\_line = '-'.join('' for x in range(100))

print(dash\_line)

print(f'INPUT PROMPT:\n{prompt}')

print(dash\_line)

print(f'BASELINE HUMAN SUMMARY:\n{summary}\n')

print(dash\_line)

print(f'MODEL GENERATION - ZERO SHOT:\n{original\_model\_text\_output}')

1. Copy the below python code. Execute the cell.

# Evaluate rouge metrics for Zero shot

original\_model\_results = rouge.compute(

predictions=original\_model\_summaries,

references=human\_baseline\_summaries[0:len(original\_model\_summaries)],

use\_aggregator=True,

use\_stemmer=True,

)

print('ORIGINAL MODEL:')

print(original\_model\_results)

1. Write a python code to find out One shot prompting for dataset index 200 and print the response

%%time

human\_baseline\_summaries = []

original\_model\_summaries = []

# One shot prompting for dataset index 200

index = 200

dialogue = dataset['test'][index]['dialogue']

summary = dataset['test'][index]['summary']

human\_baseline\_summaries.append(dataset['test'][0:10]['summary'])

# Prompt preparation

prompt = f"""

Summarize the following conversation.

#Person1#: Have you considered upgrading your system?

#Person2#: Yes, but I'm not sure what exactly I would need.

#Person1#: You could consider adding a painting program to your software. It would allow you to make up your own flyers and banners for advertising.

#Person2#: That would be a definite bonus.

#Person1#: You might also want to upgrade your hardware because it is pretty outdated now.

#Person2#: How can we do that?

#Person1#: You'd probably need a faster processor, to begin with. And you also need a more powerful hard disc, more memory and a faster modem. Do you have a CD-ROM drive?

#Person2#: No.

#Person1#: Then you might want to add a CD-ROM drive too, because most new software programs are coming out on Cds.

#Person2#: That sounds great. Thanks.

Summary:

#Person1# teaches #Person2# how to upgrade software and hardware in #Person2#'s system.

#############################

Summarize the following conversation.

{dialogue}

Summary: """

print(prompt)

payload = {

"inputs": prompt,

"parameters": {

"do\_sample": True,

"top\_p": 0.9,

"temperature": 0.1,

"max\_new\_tokens": 1024,

"stop": ["<|endoftext|>", "<\/s>"]

}

}

# predict the response

response = predictor.predict(payload)

print(response[0]["generated\_text"])

original\_model\_text\_output = response[0]["generated\_text"]

original\_model\_summaries.append(original\_model\_text\_output)

1. Write a python code to print input prompt, human summary, and model generated for One shot

# print input prompt, human summary, and model generated - One shot

dash\_line = '-'.join('' for x in range(100))

print(dash\_line)

print(f'INPUT PROMPT:\n{prompt}')

print(dash\_line)

print(f'BASELINE HUMAN SUMMARY:\n{summary}\n')

print(dash\_line)

print(f'MODEL GENERATION - ONE SHOT:\n{original\_model\_text\_output}')

1. Write a python code to evaluate rouge metrics for One shot

# Evaluate rouge metrics for One shot

original\_model\_results = rouge.compute(

predictions=original\_model\_summaries,

references=human\_baseline\_summaries[0:len(original\_model\_summaries)],

use\_aggregator=True,

use\_stemmer=True,

)

print('ORIGINAL MODEL:')

print(original\_model\_results)

1. Now you have the Rouge scores for both the zero-shot and the one-shot approach. The last step is to calculate the following and submit the results:

rouge1(zeroshot)-rouge1(oneshot), rouge2(zeroshot)-rouge2(oneshot), rougeL(zeroshot)-rougeL(oneshot), rougeLsum(zeroshot)-rougeLsum(oneshot)