Model Evaluation V2:

Set Up

There are some initial steps required for setup. If you recieve warnings after running these cells, you can ignore them as they won't impact the code running in the notebook. Run the cell below to ensure you're using the latest version of the Sagemaker Python client library. Restart the Kernel after you run this cell.

In [1]:
```
!pip install ipywidgets==7.0.0 --quiet
!pip install --upgrade sagemaker datasets --quiet
```

! Restart the notebook kernel now after running the above cell and before you run any cells below !

To deploy the model on Amazon Sagemaker, we need to setup and authenticate the use of AWS services. Yo'll uuse the execution role associated with the current notebook instance as the AWS account role with SageMaker access. Validate your role is the Sagemaker IAM role you created for the project by running the next cell.

In [2]:
```
import sagemaker, boto3, json
from sagemaker.session import Session

sagemaker_session = Session()
aws_role = sagemaker_session.get_caller_identity_arn()
aws_region = boto3.Session().region_name
sess = sagemaker.Session()
print(aws_role)
print(aws_region)
print(sess)
```

```
sagemaker.config INFO - Not applying SDK defaults from location: /etc/xdg/sagemaker/config.yaml
sagemaker.config INFO - Not applying SDK defaults from location: /home/ec2-user/.config/sagemaker/config.yaml
arn:aws:iam::085871803823:role/service-role/SageMaker-ProjectSagemakerRoleV2
us-west-2
<sagemaker.session.Session object at 0x7fa92816b640>
```

2. Select Text Generation Model Meta Llama 2 7B

Run the next cell to set variables that contain the values of the name of the model we want to load and the version of the model .

In [3]:
```
(model_id, model_version,) = ("meta-textgeneration-llama-2-7b","2.*",)
```

Running the next cell deploys the model This Python code is used to deploy a machine learning model using Amazon SageMaker's JumpStart library.

1. Import the `JumpStartModel` class from the `sagemaker.jumpstart.model` module.
2. Create an instance of the `JumpStartModel` class using the `model_id` and `model_version` variables created in the previous cell. This object represents the machine learning model you want to deploy.
3. Call the `deploy` method on the `JumpStartModel` instance. This method deploys the model on Amazon SageMaker and returns a `Predictor` object.

The `Predictor` object (`predictor`) can be used to make predictions with the deployed model. The `deploy` method will automatically choose an endpoint name, instance type, and other deployment parameters. If you want to specify these parameters, you can pass them as arguments to the `deploy` method.

The next cell will take some time to run. It is deploying a large language model, and that takes time. You'll see dashes (--) while it is being deployed. Please be patient! You'll see an exclamation point at the end of the dashes (---!) when the model is deployed and then you can continue running the next cells.

You might see a warning "For forward compatibility, pin to model_version..." You can ignore this warning, just wait for the model to deploy.

In [4]:
```
from sagemaker.jumpstart.model import JumpStartModel

model = JumpStartModel(model_id=model_id, model_version=model_version, instance_type="ml.g5.2xlarge")
predictor = model.deploy()
```

```
For forward compatibility, pin to model_version='2.*' in your JumpStartModel or JumpStartEstimator definitions. Note that major
version upgrades may have different EULA acceptance terms and input/output signatures.
Using vulnerable JumpStart model 'meta-textgeneration-llama-2-7b' and version '2.1.8'.
Using model 'meta-textgeneration-llama-2-7b' with wildcard version identifier '2.*'. You can pin to version '2.1.8' for more st
able results. Note that models may have different input/output signatures after a major version upgrade.
```

```
----------------!
```

Invoke the endpoint, query and parse response

The next step is to invoke the model endpoint, send a query to the endpoint, and recieve a response from the model.

Running the next cell defines a function that will be used to parse and print the response from the model.

In [5]:
```
def print_response(payload, response):
    print(payload["inputs"])
    print(f"> {response[0]['generation']}")
    print("\n==================================\n")
```

The model takes a text string as input and predicts next words in the sequence, the input we send it is the prompt.

The prompt we send the model should relate to the domain we'd like to fine-tune the model on. This way we'll identify the model's domain knowledge before it's fine-tuned, and then we can run the same prompts on the fine-tuned model.

In [6]:
```
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 shown in Table 1.
Table 1: The results for the short in the money options
Strike Price (CAD)
Exercise Price (CAD)
Option Value (CAD)
Exercise Price (USD)
Option Value (USD)
0.

==================================
```

In [7]:
```
# Delete the SageMaker endpoint and the attached resources
predictor.delete_model()
predictor.delete_endpoint()
```

Verify your model endpoint was deleted by visiting the Sagemaker dashboard and choosing `endpoints` under 'Inference' in the left navigation menu. If you see your endpoint still there, choose the endpoint, and then under "Actions" select **Delete**

Model Fine Tuning V2:

In [1]:
```
!pip install --upgrade sagemaker datasets
```

```
Requirement already satisfied: sagemaker in /home/ec2-user/anaconda3/envs/pytorch_p310/lib/python3.10/site-packages (2.232.0)
Requirement already satisfied: datasets in /home/ec2-user/anaconda3/envs/pytorch_p310/lib/python3.10/site-packages (3.0.0)
Requirement already satisfied: attrs<24,>=23.1.0 in /home/ec2-user/anaconda3/envs/pytorch_p310/lib/python3.10/site-packages
(from sagemaker) (23.2.0)
Requirement already satisfied: boto3<2.0,>=1.34.142 in /home/ec2-user/anaconda3/envs/pytorch_p310/lib/python3.10/site-package
s (from sagemaker) (1.35.16)
Requirement already satisfied: cloudpickle==2.2.1 in /home/ec2-user/anaconda3/envs/pytorch_p310/lib/python3.10/site-packages
(from sagemaker) (2.2.1)
Requirement already satisfied: docker in /home/ec2-user/anaconda3/envs/pytorch_p310/lib/python3.10/site-packages (from sagema
ker) (7.1.0)
Requirement already satisfied: google-pasta in /home/ec2-user/anaconda3/envs/pytorch_p310/lib/python3.10/site-packages (from
sagemaker) (0.2.0)
Requirement already satisfied: importlib-metadata<7.0,>=1.4.0 in /home/ec2-user/anaconda3/envs/pytorch_p310/lib/python3.10/si
te-packages (from sagemaker) (6.11.0)
Requirement already satisfied: jsonschema in /home/ec2-user/anaconda3/envs/pytorch_p310/lib/python3.10/site-packages (from sa
gemaker) (4.23.0)
Requirement already satisfied: numpy<2.0,>=1.9.0 in /home/ec2-user/anaconda3/envs/pytorch_p310/lib/python3.10/site-packages
(from sagemaker) (1.26.4)
Requirement already satisfied: packaging>=20.0 in /home/ec2-user/anaconda3/envs/pytorch_p310/lib/python3.10/site-packages (fr
```

In [2]:
```
model_id, model_version = "meta-textgeneration-llama-2-7b", "2.*"
```

In the cell below, choose the training dataset text for the domain you've chosen and update the code in the cell below:

To create a finance domain expert model:

- `"training": f"s3://genaiwithawsproject2024/training-datasets/finance"`

To create a medical domain expert model:

- `"training": f"s3://genaiwithawsproject2024/training-datasets/medical"`

To create an IT domain expert model:

- `"training": f"s3://genaiwithawsproject2024/training-datasets/it"`

In [3]:
```
from sagemaker.jumpstart.estimator import JumpStartEstimator
import boto3

estimator = JumpStartEstimator(model_id=model_id,  environment={"accept_eula": "true"},instance_type = "ml.g5.2xlarge")

estimator.set_hyperparameters(instruction_tuned="False", epoch="5")

#Fill in the code below with the dataset you want to use from above
#example: estimator.fit({"training": f"s3://genaiwithawsproject2024/training-datasets/finance"})
estimator.fit({ "training": f"s3://genaiwithawsproject2024/training-datasets/finance" })
```

```
INFO:root:Key: avg_epoch_time, Value: 7.648085441599994
INFO:root:Key: avg_checkpoint_time, Value: 0.7353730877999851
INFO:root:Combining pre-trained base model with the PEFT adapter module.
Loading checkpoint shards:   0%|          | 0/2 [00:00<?, ?it/s]
Loading checkpoint shards:  50%|█████     | 1/2 [00:29<00:29, 29.64s/it]
Loading checkpoint shards: 100%|██████████| 2/2 [00:37<00:00, 16.94s/it]
Loading checkpoint shards: 100%|██████████| 2/2 [00:37<00:00, 18.85s/it]
INFO:root:Saving the combined model in safetensors format.
INFO:root:Saving complete.
INFO:root:Copying tokenizer to the output directory.
INFO:root:Putting inference code with the fine-tuned model directory.
2024-09-15 10:12:01,385 sagemaker-training-toolkit INFO     Waiting for the process to finish and give a return code.
2024-09-15 10:12:01,385 sagemaker-training-toolkit INFO     Done waiting for a return code. Received 0 from exiting process.
2024-09-15 10:12:01,385 sagemaker-training-toolkit INFO     Reporting training SUCCESS

2024-09-15 10:12:07 Uploading - Uploading generated training model
2024-09-15 10:12:50 Completed - Training job completed
Training seconds: 692
Billable seconds: 692
```

In [4]:
```
'''
# Do not use estimator.deploy() without mentioning the instance_type.
# It's because when you call estimator.deploy() without explicitly setting the instance_type for the endpoint,
# SageMaker selects a default instance type for hosting, which, in this case, is ml.g5.12xlarge.
# However, Udacity doesn't allow instance type more than "ml.*.2xlarge".
'''
finetuned_predictor = estimator.deploy(instance_type="ml.g5.2xlarge", initial_instance_count=1)
```

```
INFO:sagemaker:Creating model with name: meta-textgeneration-llama-2-7b-2024-09-15-10-13-03-014
INFO:sagemaker:Creating endpoint-config with name meta-textgeneration-llama-2-7b-2024-09-15-10-13-03-011
INFO:sagemaker:Creating endpoint with name meta-textgeneration-llama-2-7b-2024-09-15-10-13-03-011

----------!
```

In [5]:
```
def print_response(payload, response):
    print(payload["inputs"])
    print(f"> {response}")
    print("\n==================================\n")
```

In [6]:
```
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': ' were very impressive. The median result was 37% and the average result was 41%. The results for the out of the money options were very disappointing. The median result was -13% and the average result was -18%.\nThe results for the long in the money options were'}]

==================================
```

In [7]:
```
finetuned_predictor.delete_model()
finetuned_predictor.delete_endpoint()
```

```
INFO:sagemaker:Deleting model with name: meta-textgeneration-llama-2-7b-2024-09-15-10-13-03-014
INFO:sagemaker:Deleting endpoint configuration with name: meta-textgeneration-llama-2-7b-2024-09-15-10-13-03-011
INFO:sagemaker:Deleting endpoint with name: meta-textgeneration-llama-2-7b-2024-09-15-10-13-03-011
```