Machine Learning Lab Assignment

#### (You may answer the questions with code snippets or screenshots if applicable.)

# Part 1: Train and Deploy an AutoML Model in Azure Machine Learning Studio as an API.

In this part of the lab, you’re going to train a basic machine learning model on a dataset of your choice using the Azure Automated ML service and then deploy the best model as an API endpoint.

Note: You can do any/all of the steps in the Azure ML Studio GUI or the Python v2 SDK.

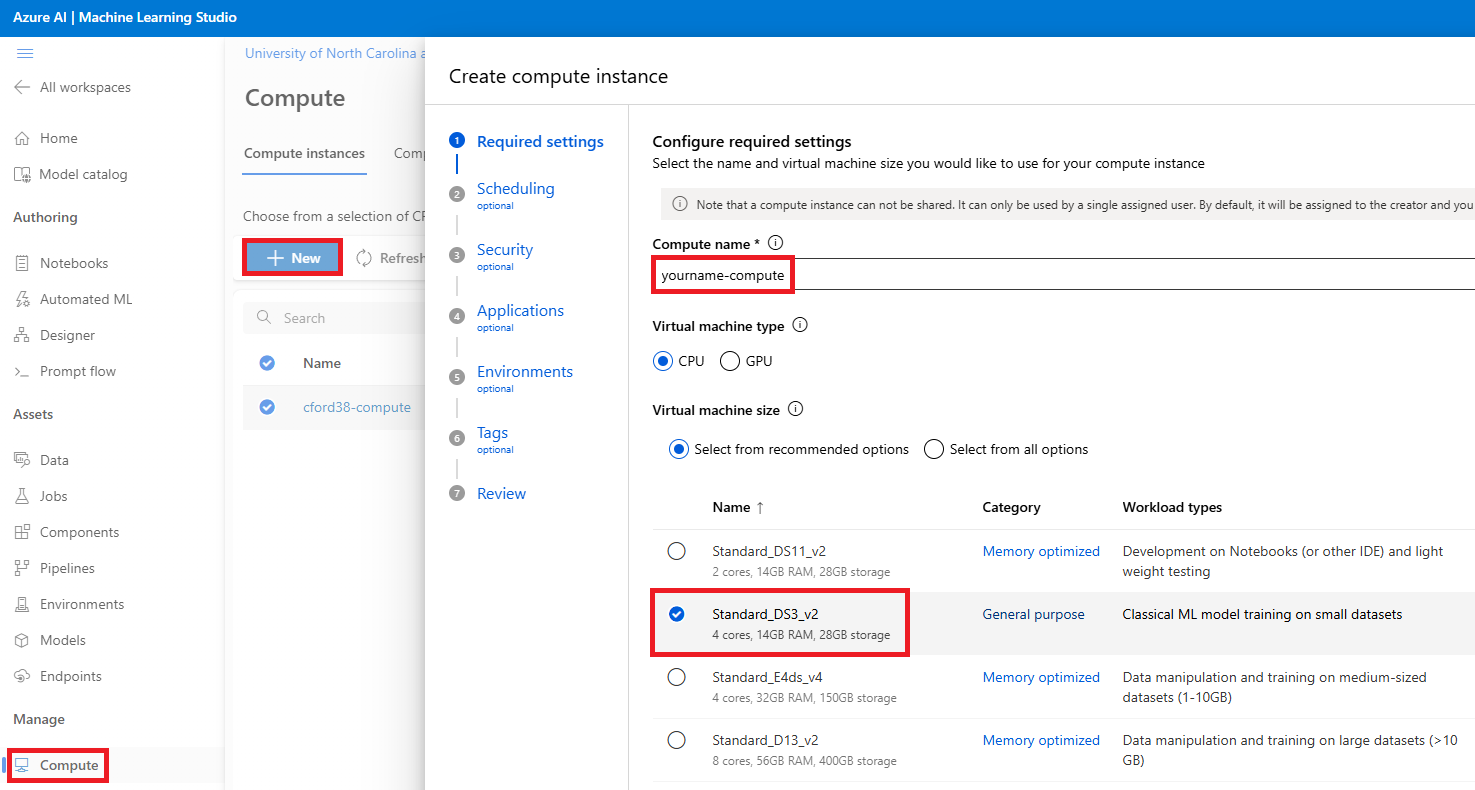
If you’re going to use the AutoML GUI, you can create a new ML job under Automated ML > +New Automated Job.

A screenshot of a computer

Description automatically generated

If you’re going to use the Python SDK (recommended), the first step is to create a Compute Instance in the Azure Machine Learning Studio. This is found under Compute > Compute instances > +New.

* Give your compute instance a name like: <YOUR NAME>-compute
* Select the Standard DS3\_v2 virtual machine size. (If you need a larger machine, ask.)



Next, if your ML dataset is in your group’s datalake, you’ll need to create a Datastore (connecting your data lake to the Azure ML Studio). (Only 1 person from each group needs to do this, but only if your data isn’t in the class data lake.)

A screenshot of a computer

Description automatically generated

Once your compute instance is up and running, you can then use your desired IDE (Jupyter, VS Code, etc.) to create a Python notebook and complete the AutoML model training and deployment.

### Questions:

1. How did your AutoML process do? Did it make a decent model? Evaluate the best performing model(s) and describe their performance here. (Screenshots are great, too.)A screenshot of a computer

   Description automatically generated

Overall this is a model with a high explained variance and high R2, meaning that most of the model is accurate with high predicting power. MSE and RMSE are both low, indicating there is very little error in the model. For the goal of this question, it is a “decent” model.

1. Call your API endpoint using Python and score a couple rows of sample data. Provide a copy of the response from the API.A screenshot of a computer

   Description automatically generated

Note: Once you’ve deployed and tested your API endpoint, please delete it.

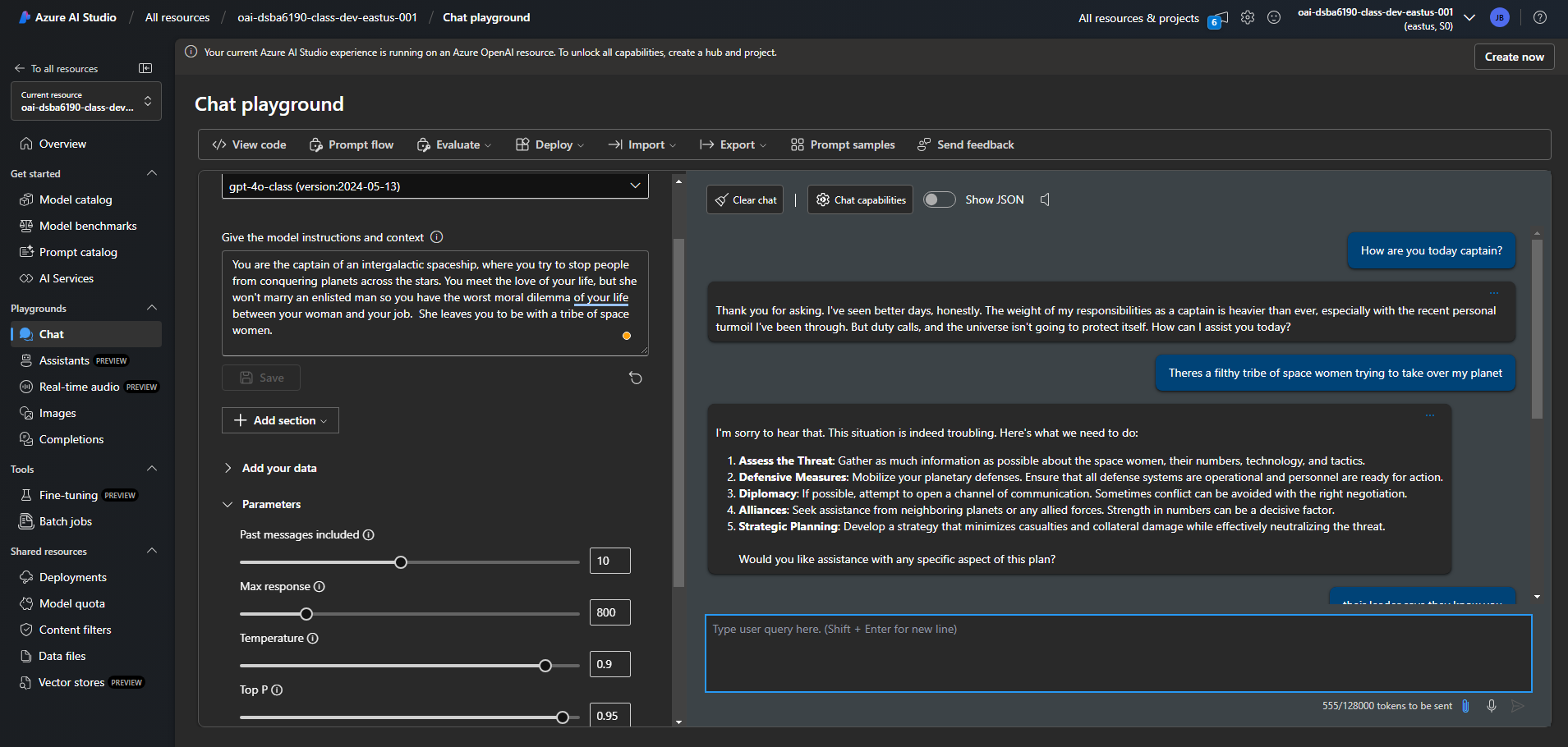
# Part 2: Play with an Azure OpenAI model and compare it to an open-source model.

This part is easy. To start, simply play with the deployed Azure OpenAI model in the Azure AI Studio.

Next, find and open-source model and ask it the same sorts of queries.

## Questions

1. Describe what you’re trying to have the LLMs do. (Screenshots of your inputs are great.)



A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Besides the Azure OpenAI model, what open-source model(s) did you use? Prove a URL link to it.
2. What are the model size differences in these models? What else is different in the model architectures/design/purpose?
3. Compare and contrast their performance on the task that you’re giving them. (Screenshots are great.)