Guided Lab: Amazon SageMaker and Importing Data

Lab overview

In this lab, you will learn how to launch an Amazon SageMaker notebook instance. From that instance, you will learn how to create a Jupyter notebook. You will learn how to create code and Markdown cells within the notebook. You will download data from an external source, then learn how to save your notebook locally so you can continue working on labs across sessions.

Amazon SageMaker is a fully managed service that provides every developer and data scientist with the ability to build, train, and deploy machine learning (ML) models quickly. Amazon SageMaker removes the heavy lifting from each step of the machine learning (ML) process to make it easier to develop high quality models.

Objectives

After completing this lab, you will be able to:

- Launch an Amazon SageMaker notebook instance
- Launch a Jupyter notebook
- Run code in a notebook
- Download data from an external source
- Upload and download a Jupyter notebook to your local machine

Prerequisites

This lab requires:

- Access to a notebook computer with Wi-Fi and Microsoft Windows, macOS, or Linux (Ubuntu, SUSE, or Red Hat)
- For Microsoft Windows users: Administrator access to the computer
- An internet browser such as Chrome, Firefox, or IE9 (previous versions of Internet Explorer are not supported)

Duration

This lab takes approximately **30 minutes**. The lab will remain active for **180 minutes**

AWS services not used in this lab

In the lab environment, AWS services that are not used in this lab are disabled. In addition, the capabilities of the services that are used in this lab are limited to what the lab requires. Expect errors when you access other services or perform actions beyond those provided in this lab.

Accessing the AWS Management Console

1. At the top of these instructions, choose `Start Lab` to launch your lab.
   A **Start Lab** panel opens, which displays the lab status.

2. Wait until you see the message *Lab status: ready*, then close the **Start Lab** panel by choosing the **X**.
3. At the top of these instructions, choose `AWS`
   This will open the AWS Management Console in a new browser tab. The system will automatically log you in.
   Tip: If a new browser tab does not open, there will typically be a banner or icon at the top of your browser that indicates that your browser is preventing the website from opening pop-up windows. Select the banner or icon and then choose **Allow pop ups**.
4. Arrange the **AWS Management Console** browser tab so that it displays next to these instructions. Ideally, you should be able to see both browser tabs at the same time, which can make it easier to follow the lab steps.

Task 1: Creating a Jupyter notebook with Amazon SageMaker

In this task, you will access Amazon SageMaker in the AWS Management Console and create a Jupyter notebook.

To create a Jupyter notebook with Amazon SageMaker:

5. On the AWS Management Console, on the **Services** menu, choose **Amazon SageMaker**.
6. From the navigation menu on the left, expand the **Notebook** section and choose **Notebook instances**.
7. Choose **Create notebook instance**.
8. In the **Notebook instance name** box, enter `Mynotebook`
9. From the **Notebook instance type** dropdown list, choose **ml.m4.xlarge**.
10. Now set the Platform identifier type to **notebook-al2-v1**, this ensures you are using Amazon Linux 2 as operating system type for the notebook instance. To learn more about notebook instances and the operating system type options, please refer to Amazon Linux 2 vs Amazon Linux notebook instances.
11. Under **Additional configuration**, choose the dropdown menu for **Lifecycle configuration** and choose the lifecycle configuration that contains *ml-pipeline*.

This will automatically add the correct Jupyter Notebooks to your Amazon SageMaker Notebook instance.

12. Leave the remaining settings at their default values.
13. Choose **Create notebook instance**.
    Your new notebook is in the **Notebook instances** list with a status of *Pending*. When the status changes to *InService*, you can continue to the next step. While you are waiting for the status to change, you can read more about Jupyter notebooks in either of the following references:
    - Jupyter.org
    - Jupyter Notebook Tutorial
14. It will take 2-5 minutes for the Notebook instance to start.
15. At the end of the row, choose **Open JupyterLab** to open the Jupyter notebook instance.

Task 2: Introducing JupyterLab

In this task, you will explore the JupyterLab GUI on Amazon SageMaker and work through a sample Jupyter notebook. Jupyter notebooks enable you to create and share documents that contain both code and rich text elements, such as equations.

15. In the JupyterLab GUI, open the folder for your desired language, then open the **PythonCheatSheeet.ipynb** notebook file (which was automatically uploaded to the JupyterLab IDE). The notebook will open in a new tab in the editor.
    The JupyterLab environment will look like the following image. You will now examine at its components in detail.

    The top includes the menu bar, which provides all the commands to control the notebook and environment. It includes:
    - **File** – Used to save, revert, or create a checkpoint for your notebook.
    - **Edit** – Edit your notebook structure, which includes the following commands.
        - Cut (Shortcut: x)
        - Copy (Shortcut: c)
        - Paste (Shortcut for *Paste below*: V)
        - Delete (Shortcut: dd)
        - Undo delete (Shortcut: z)
    - **View** – Toggles different options for your notebook.

   - **Run** – Commands to run the cells in the notebook.
   - **Kernel** – Change the state of the kernel (or programming language) that's used by the notebooks. Amazon SageMaker includes various kernels based on Python, depending on frameworks you want to use (for example, TensorFlow, Pytorch, and MXNet).
   - **Git** – Commands to interact with a git repository (if you have one that's configured to work with JupyterLab).
   - **Tabs** – Commands to navigate the notebook tabs.
   - **Settings** – Commands to change your JupyterLab preferences.
   - **Help** – Information to help you use JupyterLab.

16. The left side of the screen (*1*) includes the navigation sidebar and the main work area (*2*). The navigation sidebar contains the following tabs:
    - – File browser
    - – List of running kernels and terminals
    - - Git repository viewer
    - – JupyterLab command palette
    - – Notebook tools
    - – View the open tabs within the environment
    - – View the headings of the notebook
    - – List of SageMaker samples

17. The main work area in JupyterLab enables you to arrange documents (such as notebooks, text files, and others) and other activities (such as terminals, code consoles, and others) into panels of tabs that can be resized or subdivided:
    - Drag a tab to the center of a tab panel to move the tab to the panel.
    - Subdivide a tab panel by dragging a tab to the left, right, top, or bottom of the pane.
18. The work area has a single current activity. The tab for the current activity is marked with a colored top border (blue by default).
    The main work area has a toolbar, which changes depending on the type of file that's open and in focus. The toolbar for a Jupyter notebook looks like the following example:

    1 – Save, insert, cut, copy, and paste.
    2 – Run a cell, interrupt a cell, and restart the kernel button.
    3 – A dropdown menu for changing the cell type. The run behavior of a cell is determined by the cell's type. Cells include the following types.
    - **Code cells:** The cells that take input in the form of code.
    - **Markdown cells:** Cells that contain additional information as text that's in a language called *Markdown*. The output of this cell is text that's formatted similar to an HTML page. You will look at this cell in the next task step.
    - **Raw NBconvert cells:** Raw cells that don't give you an output.
19. **4** – The kernel name and its running status. Solid indicates that code is being run.
    You can run code in the cell:
    - On the keyboard by pressing SHIFT + ENTER
    - In the toolbar by choosing **| Run**
    - In the menu bar by choosing **Cell > Run Cells**
20. The first cell in the notebook is a *Markdown cell*. A Markdown cell uses *Markdown*, which is a markup language with plaintext-formatting syntax. It's often used to format readme files to create rich text and additional documentation in the Jupyter notebook. You will now examine the Markdown cell.
21. To look at how to structure a Markdown cell, double-click the cell to inspect it.
    For more information about how to structure Markdown, see the Jupyter notebook Markdown documentation
22. Lastly, you have a *code cell* that you can use to run your code.
    A notebook can run code in different languages, but it can only use one kernel at a time. A code cell enables you to write code in a single block with full-syntax highlighting. Each cell is run individually, and the order can be found by using the

number to the left of the cell `In[<number>]`. Similarly, the output of that cell is given by `Out[<number>]`.

23. Run through this notebook one time, and then proceed to the next task.

Learn more To learn more about Jupyter notebooks, see the Jupyter notebook documentation.

Task 3: Opening a sample notebook

In this task, you will browse through a provided sample notebook and familiarize yourself with working in a Jupyter notebook.

19. In your JupyterLab environment, switch the left navigation view to the *Amazon SageMaker Samples*.
20. Locate the sample named `linear_learner_mnist.ipynb`. Choose the sample to load it into the main work area.
21. To create a copy, in the file header, choose the **Create a Copy** button and in the dialog box, choose **Create a Copy** again.
    The sample notebook will be copied to your JupyterLab environment.
    Feel free to scroll through the notebook. However, be aware that you won't be able to run the steps because an Amazon Simple Storage Service (Amazon S3) bucket is required.

Task 4: Importing data

In this task, you will create a new notebook for your labs and add code to the notebook that will download and extract the data that's used throughout the remainder of the guided labs. For this task, you will enter code into the notebook. However, in future tasks, you will work through an existing notebook so you can save time by not typing.

The data you will use throughout the guided labs is a *dataset that contains medical data about the vertebral column*. To learn more about this dataset, see the Vertebral Column Dataset at the UC Irvine Machine Learning Repository.

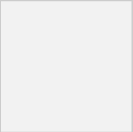
To obtain, extract, and load the data into a pandas DataFrame:

22. In your JupyterLab environment, choose **File > New** and then choose **Notebook**.

23. In the **Select kernel** dialog window, choose `conda_python3` and then choose **Select**.
    You should have a blank code cell in the main work area. You will change this blank cell to a Markup cell and add some text.
24. On the keyboard, press M. The cell type in the dropdown menu should change to *Markdown*. If not, use the dropdown menu to change the cell type.

Next, you will give your notebook a title. Remember that a notebook can contain both code and Markdown.

25. Press ENTER and in the cell, enter the following text:
    ```
    # Importing the data
    ```
26. To exit the cell and render the Markup, press SHIFT + ENTER.
    Next, you must download the data from the internet. You will use a URL, but the file is a .zip file. You could download the .zip file and extract it. However, because the file is small, you can download the file to a stream. You will then extract the files so you can save it locally.
    Start by adding a few imports.
27. On the keyboard, press ENTER and then type the following code into the cell:
28.
    ```
    import warnings, requests, zipfile, io
    warnings.simplefilter('ignore')
    import pandas as pd
    from scipy.io import arff
    ```
29. Exit the cell without running the code by pressing ESC.
    Next, add the code to download and extract the .zip file.
30. Add a cell below by pressing B.
31. Press ENTER and in the cell, enter the following code:
32.
    ```
    f_zip = 
    'http://archive.ics.uci.edu/ml/machine-learning-databases/00212/vertebral_column_data.zip'
    r = requests.get(f_zip, stream=True)
    Vertebral_zip = zipfile.ZipFile(io.BytesIO(r.content))
    Vertebral_zip.extractall()
    ```
33. Exit the cell without running the code by pressing ESC.
34. Select the two code cells by holding SHIFT and choosing both cells. Both cells should be highlighted, as shown in the following example.



Run the code in the two cells by pressing SHIFT + ENTER. If you have errors because of spelling errors, fix them now.

In the file viewer in the left navigation pane, you should see four new files:

- column_2C_weka.arff
- column_2C.dat
- column_3C_weka.arff

35. - column_3C.dat
36. Load the files into the main working window by clicking each file. Examine the data in these files. What format is the data in? Does the data have column headings?
37. You will now load the data from the two classes file. Switch back to your notebook, select the empty cell, press ENTER, and enter the following code:
38.

```
data = arff.loadarff('column_2C_weka.arff')
df = pd.DataFrame(data[0])
df.head()
```

39. This time, run the cell by pressing SHIFT + ENTER.

Task 5: Downloading your notebook and saving your work (Optional)

When you are finished with the lab environment—or if the time allocated to the lab environment expires—your SageMaker instance will be terminated, and you will lose any notebooks you created. If the lab requires you to save your notebook, the instructions will clearly indicate that you must do so. If you want to save your notebook yourself, you can use these instructions.

37. In the file browser in the left navigation, right-click the file that you want to save, then choose **Download**.
38. Next, choose the location where you want to save your file on your local computer.
39. If you want to continue working on the notebook, start the lab environment again, create your notebook instance, and upload the notebook you downloaded.

Conclusion

You now have successfully:

- Launched an Amazon SageMaker notebook instance
- Launched a Jupyter notebook
- Ran code in a notebook
- Downloaded and extracted data from an external source

Lab complete

Congratulations! You have completed the lab.

40. To confirm that you want to end the lab, at the top of this page, choose End Lab, and then choose **Yes**.

A panel should appear with this message: *DELETE has been initiated... You may close this message box now.*

41. To close the panel, choose the **X** in the top-right corner.