

Challenge 2: Load Sample Data and Databricks Notebooks



Duration: 60 minutes

In this exercise, you will implement a classification experiment. You will load the training data from your local machine into a dataset. Then, you will explore the data to identify the primary components you should use for prediction and use two different algorithms for predicting the classification. You will then evaluate the performance of both algorithms and choose the algorithm that performs best. The model selected will be exposed as a web service integrated with the optional sample web app at the end.

Task 1: Upload the Sample Datasets

1. Before you begin working with machine learning services, there are three datasets you need to load.
2. Download the three CSV sample datasets from here: <http://bit.ly/2wGAqrl> (If you get an error, or the page won't open, try pasting the URL into a new browser window and verify the case sensitive URL is exactly as shown). If you are still having trouble, a zip file called AdventureWorksTravelDatasets.zip is included in the lab-files folders.
3. Extract the ZIP and verify you have the following files:
 - FlightDelaysWithAirportCodes.csv
 - FlightWeatherWithAirportCode.csv
 - AirportCodeLocationLookupClean.csv
4. Open your Azure Databricks workspace. Before continuing to the next step, verify that your new cluster is running. Do this by navigating to **Compute (1)** on the left-hand menu and ensuring that the state of your cluster is **Running (2)**.

The screenshot shows the Databricks Compute interface. On the left sidebar, the 'Compute' icon is highlighted with a red box and a red circle with the number 1. The main content area has tabs for 'All-Purpose Clusters', 'Job Clusters', 'Pools', and 'Cluster Policies'. The 'All-Purpose Clusters' tab is active. Below the tabs is a '+ Create Cluster' button. A table lists the clusters:

	Name	State	Nodes
 	lab	Running	2

The 'State' column for the 'lab' cluster is highlighted with a red box and a red circle with the number 2.

5. Select **Data** (1) from the menu. Next, select **default** (2) under Databases (if this does not appear, start your cluster). Finally, select **Create Table** (3) above the Tables header.

The screenshot shows the Databricks Data interface. On the left sidebar, the 'Data' icon is highlighted with a red box and a red circle with the number 1. The main content area has a header 'Data' and a sub-header 'Databases'. Below the sub-header is a search bar 'Filter Databases'. A list of databases is shown, with 'default' highlighted by a red box and a red circle with the number 2. To the right of the 'Databases' section is a 'Tables' section with the text 'No tables'. Above the 'Tables' section is a 'Create Table' button, which is highlighted with a red box and a red circle with the number 3.

6. Select **Upload File (1)** under Create New Table, and then select either select or drag-and-drop the FlightDelaysWithAirportCodes.csv file into the file area (2). Select **Create Table with UI (3)**.

Create New Table

Data source ?

Upload File

DBFS

Other Data Sources

Partner Integrations

DBFS Target Directory ?

/FileStore/tables/ (optional)

Select

Files uploaded to DBFS are accessible by everyone who has access to this workspace. [Learn more](#)

Files ?

FlightDelaysWithAirportCodes.csv

0.4 GB

[Remove file](#)

✓ File uploaded to /FileStore/tables/FlightDelaysWithAirportCodes.csv

Create Table with UI

[Create Table in Notebook](#)

?

7. Select your cluster (1) to preview the table, then select **Preview Table (2)**.
8. Change the Table Name to `flight_delays_with_airport_codes` (3) and select the checkmark for **First row is header (4)**. Select **Create Table (5)**.

Select a Cluster to Preview the Table


Choose a cluster with which you will read and preview the data.


Cluster  lab 1

Preview Table 2

Specify Table Attributes


Specify the Table Name, Database and Schema to add this to the data UI for other users to access

Table Name  flight_delays_with_airport_c 3

Create in Database  default

File Type  CSV

Column Delimiter  ,

☒ First row is header  4

☐ Infer schema 

☐ Multi-line 

Create Table 5


 Create Table in Notebook


Table Preview

Year	Month	DayofMonth	DayOfWeek	Carrier
STRING	STRING	STRING	STRING	STRIN
2013	4	19	5	DL
2013	4	19	5	DL
2013	4	19	5	DL
2013	4	19	5	DL
2013	4	19	5	DL
2013	4	19	5	DL

9. Repeat steps 5 through 8 for the FlightWeatherWithAirportCode.csv and AirportCodeLocationLookupClean.csv files, setting the name for each dataset in a similar fashion:

- flightweatherwithairportcode_csv renamed to **flight_weather_with_airport_code**
- airportcodelocationlookupclean_csv renamed to **airport_code_location_lookup_clean**


Data

Create Table 

Databases





 Filter Databases

 default

Tables

 Filter Tables

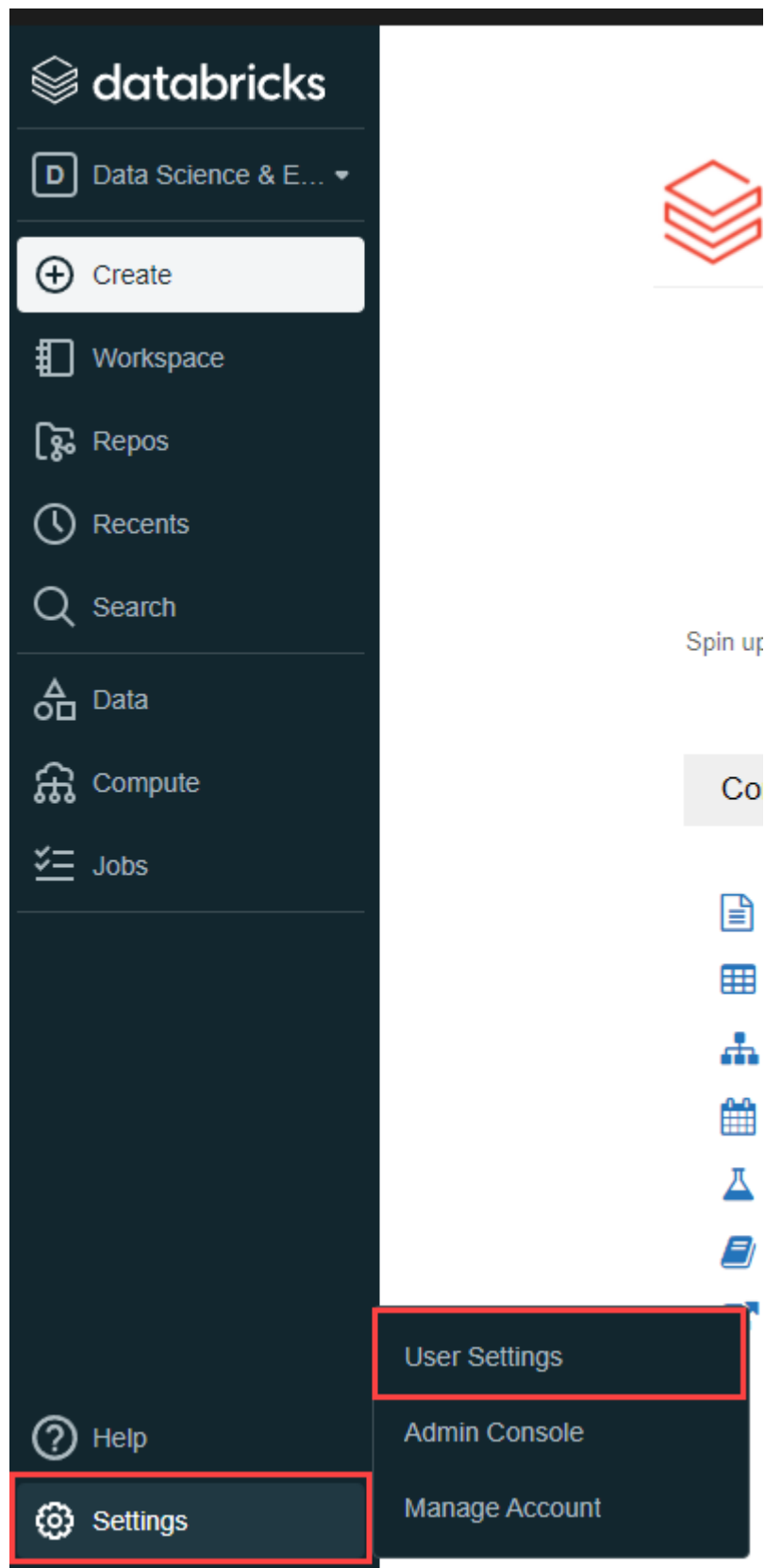
 airport_code_location_lookup_clean ▼

 flight_delays_with_airport_codes ▼

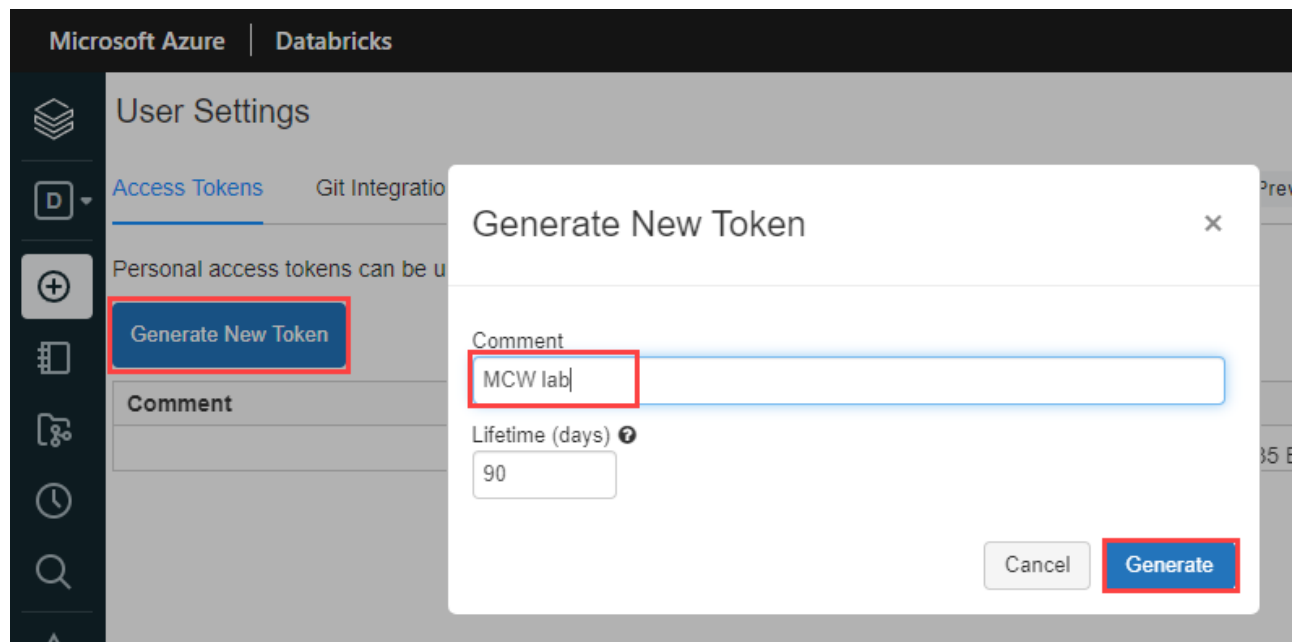
 flight_weather_with_airport_code ▼

Task 2: Open Azure Databricks and complete lab notebooks

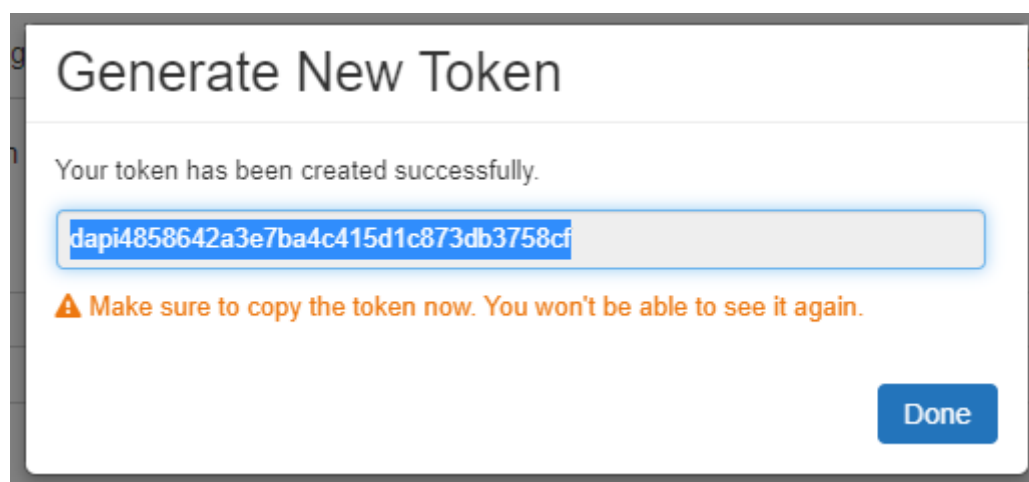
1. In Azure Databricks, select the **Settings** menu in the bottom left corner of the window, then select **User Settings**.



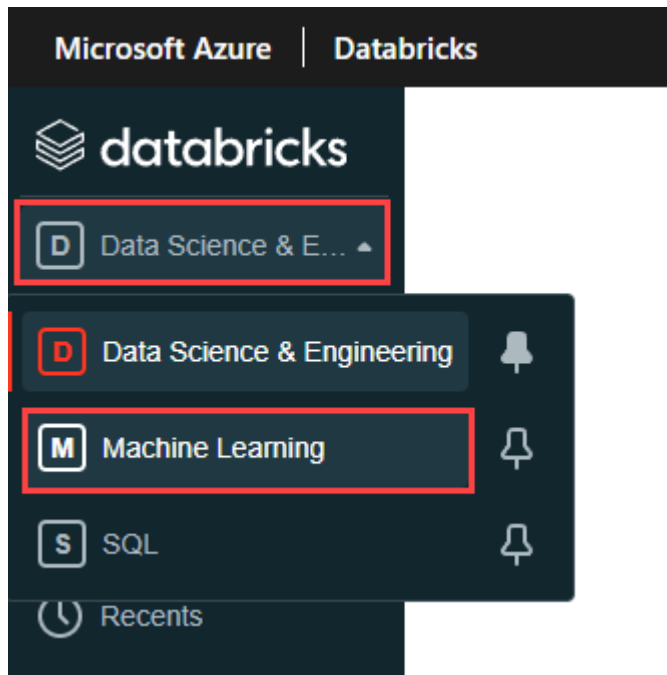
2. Select **Generate New Token** under the Access Tokens tab. Enter **MCW lab** for the comment and leave the lifetime at 90 days. Select **Generate** to generate a Personal Access Token, or PAT.



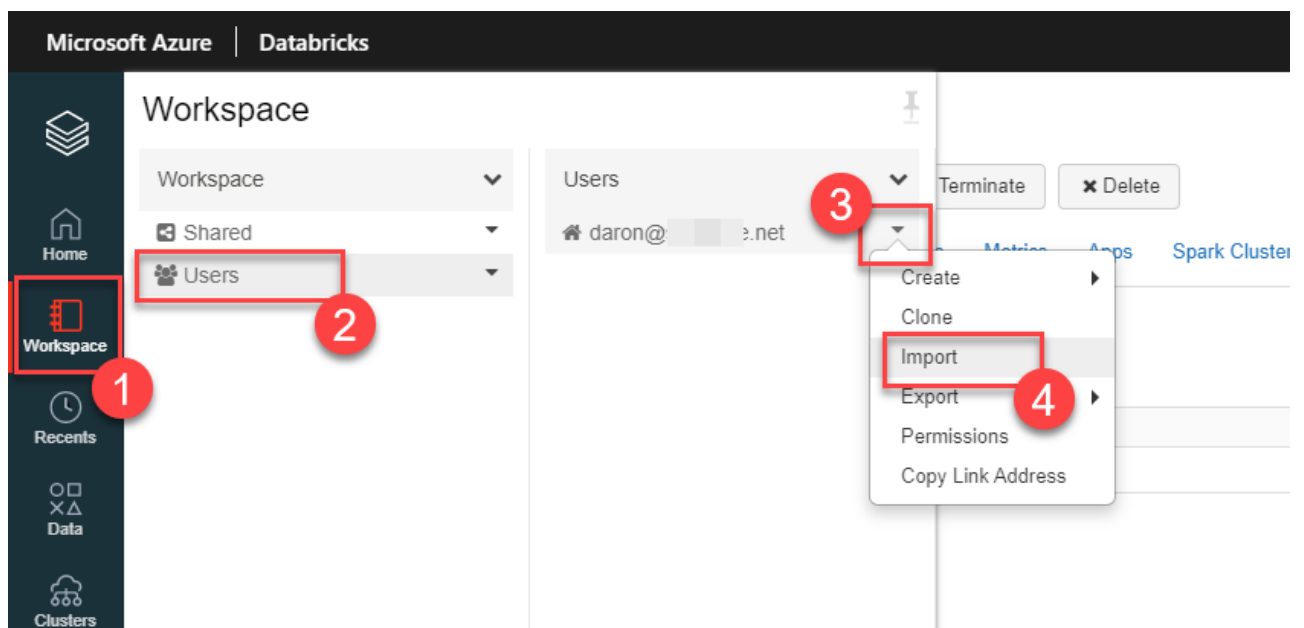
3. **Copy** the generated token and **paste it into a text editor** such as Notepad for use later in this exercise as well as in future exercises. Select **Done** once you are finished.



4. Within Azure Databricks, select **Data Science & Engineering** and choose **Machine Learning** from the list. You will need to be in this view before completing one of the notebooks later in this exercise.



5. Within Azure Databricks, select **Workspace** (1) on the menu, then **Users** (2), then select the down arrow next to your username (3). Select **Import** (4).



6. Within the Import Notebooks dialog, select Import from: **URL** (1), then paste the following into the URL textbox (2): <https://github.com/microsoft/MCW-Big-data-analytics-and-visualization/blob/main/Hands-on%20lab/lab-files/BigDataVis.dbc?raw=true>. Select **Import** (3) to continue.

Import Notebooks

Import from: ☐ File ☒ URL

<https://github.com/microsoft/MCW-Big-data-and-visualization/blob/master/BigDataVis>

Accepted formats: .dbc, .scala, .py, .sql, .r, .ipynb, .Rmd, .html

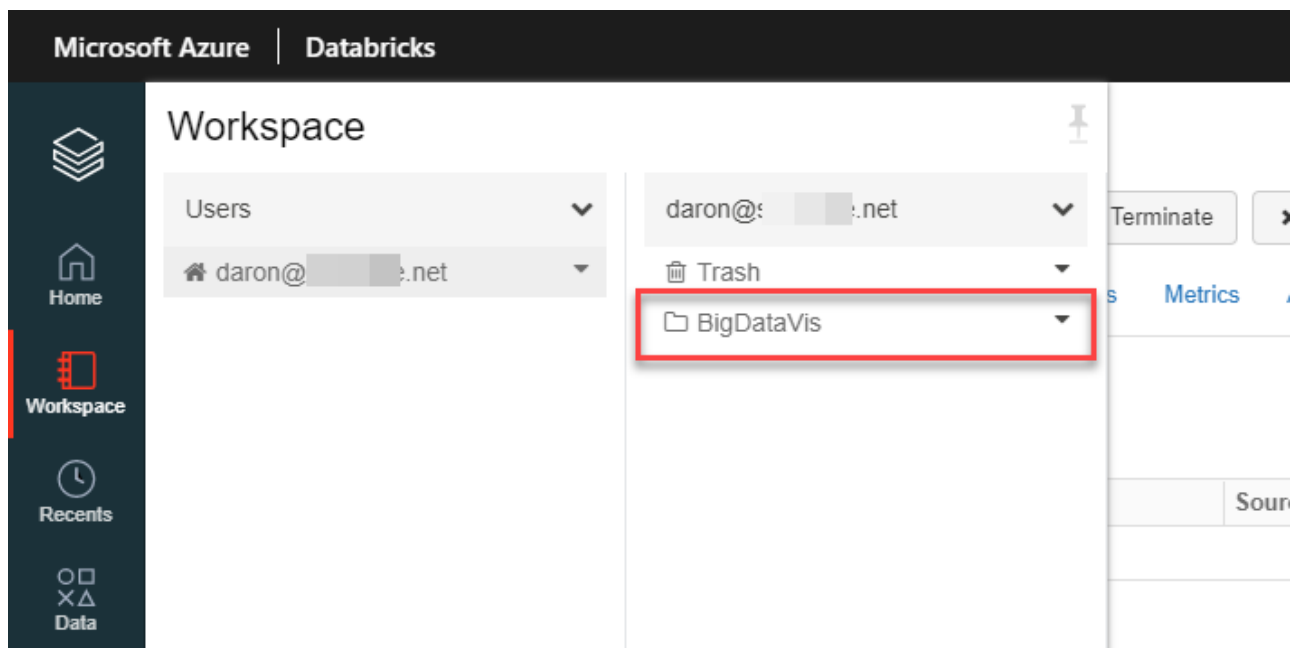
(To import a library, such as a jar or egg, [click here](#))

Cancel

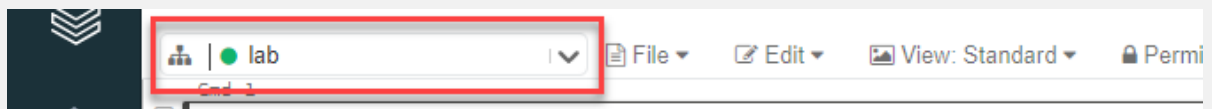
Import

Note: This Databricks archive is available within the [Hands-on lab\lab-files](#) directory of this repository. In the [BigDataVis](#) subfolder, you can also see the individual notebooks as separate files in .ipynb format.

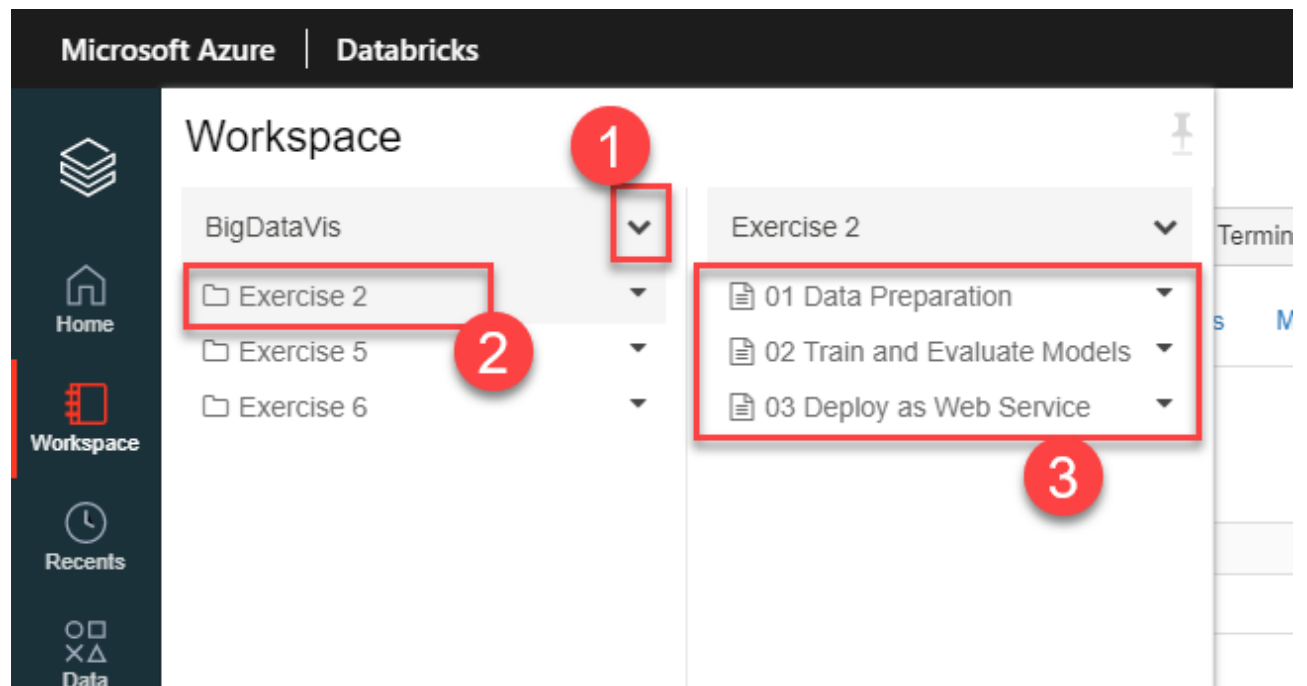
7. After importing, expand the new **BigDataVis** folder.



WARNING: When you open a notebook, make sure you attach your cluster to the notebook using the **Attach to cluster** dropdown. You will need to do this for each notebook you open.



8. Run each cell (except [Clean up](#) section in Notebook 3) of the notebooks located in the **Exercise 2** folder (01, 02 and 03) individually by selecting within the cell, then entering **Ctrl+Enter** on your keyboard. Pay close attention to the instructions within the notebook, so you understand each step of the data preparation process.



9. Do NOT run any notebooks within the Exercise 5 or 6 folders. They will be discussed later in the lab.