

Developing Big Data Solutions with Azure Machine Learning

Lab 1 - Getting Started with Azure Machine Learning

Overview

In this lab, you will provision Azure Machine Learning workspace and use it to explore data from big data sources.

What You'll Need

To complete the labs, you will need the following:

- A Microsoft account
- A Microsoft Azure subscription
- A Windows, Linux, or Mac OS X computer
- Azure Storage Explorer
- The lab files for this course

Note: To set up the required environment for the lab, follow the instructions in the [setup guide](#) for this course.

Exercise 1: Provision an Azure Machine Learning Workspace

In this exercise, you will create an Azure Machine Learning workspace so that you can experiment with data.

Note: The Microsoft Azure portal is continually improved in response to customer feedback. The steps in this exercise reflect the user interface of the Microsoft Azure portal at the time of writing but may not match the latest design of the portal exactly.

Create an Azure Machine Learning Workspace and Plan

Before you can use Azure Machine Learning, you must provision a workspace. In this case, you will provision the workspace in your Azure subscription (note that you can also provision a free Azure machine learning workspace that does not require an Azure subscription – free workspaces are subject to some restrictions).

1. In a web browser, navigate to <http://portal.azure.com>, and if prompted, sign in using the Microsoft account that is associated with your Azure subscription.

2. In the Microsoft Azure portal, click **New**. Then in the **Internet of Things** menu, click **Machine Learning Workspace** (be careful not to select *Machine Learning Web Service*).
3. In the **Machine Learning workspace** blade, enter the following settings, and then click **Create**:
 - **Workspace name**: *Enter a unique name.*
 - **Subscription**: *Select your Azure subscription*
 - **Resource Group**: *Create a new resource group with a unique name.*
 - **Location**: *Select any available region.*
 - **Storage account**: *Create a new storage account with a unique name.*
 - **Workspace pricing tier**: Standard
 - **Web service plan**: *Create a new web service plan with a unique name.*
 - **Web service plan pricing tier**: DEVTEST Standard
 - **Pin to dashboard**: *Not selected*
4. In the Azure portal, view **Notifications** to verify that deployment has started. Then wait for the workspace to be deployed (this can take a few minutes.)
5. Click **All resources** and verify that your subscription now includes the following new resources:
 - A Machine Learning Workspace.
 - A Machine Learning Plan.
 - A Storage Account.

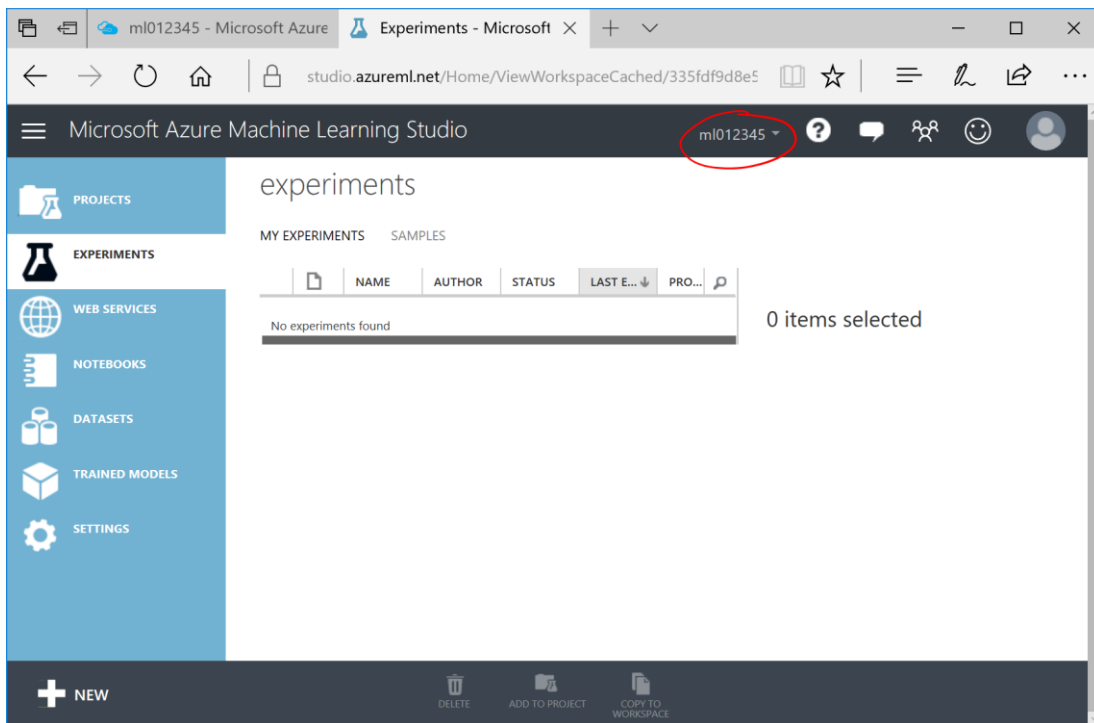
Exercise 2: Explore Data with Azure Machine Learning

In this exercise, you will use Azure Machine Learning Studio to explore data gathered from patients in a health clinic that are participating in a study on the effects of physical exercise; and find relationships between physiology, exercise duration, and calories burned. The goal of the exercise is to familiarize yourself with Azure Machine Learning Studio.

Open Azure Machine Learning Studio

Now that you have a workspace, you can use Azure Machine Learning Studio to work with data.

1. In the Azure portal, browse to the workspace you created in the previous procedure.
2. In the blade for your workspace, click **Launch Machine Learning Studio**. This opens a new browser page.
3. In the new browser page, sign into Azure Machine Learning Studio using the Microsoft account associated with your Azure subscription.
4. In Azure Machine Learning Studio, at the top right, ensure that the name of the workspace you created in the previous procedure is displayed as shown below:



Note: If a different workspace name is displayed, you may already have some workspaces associated with your account – in which case select your new workspace in the drop-down list.

Upload a Dataset

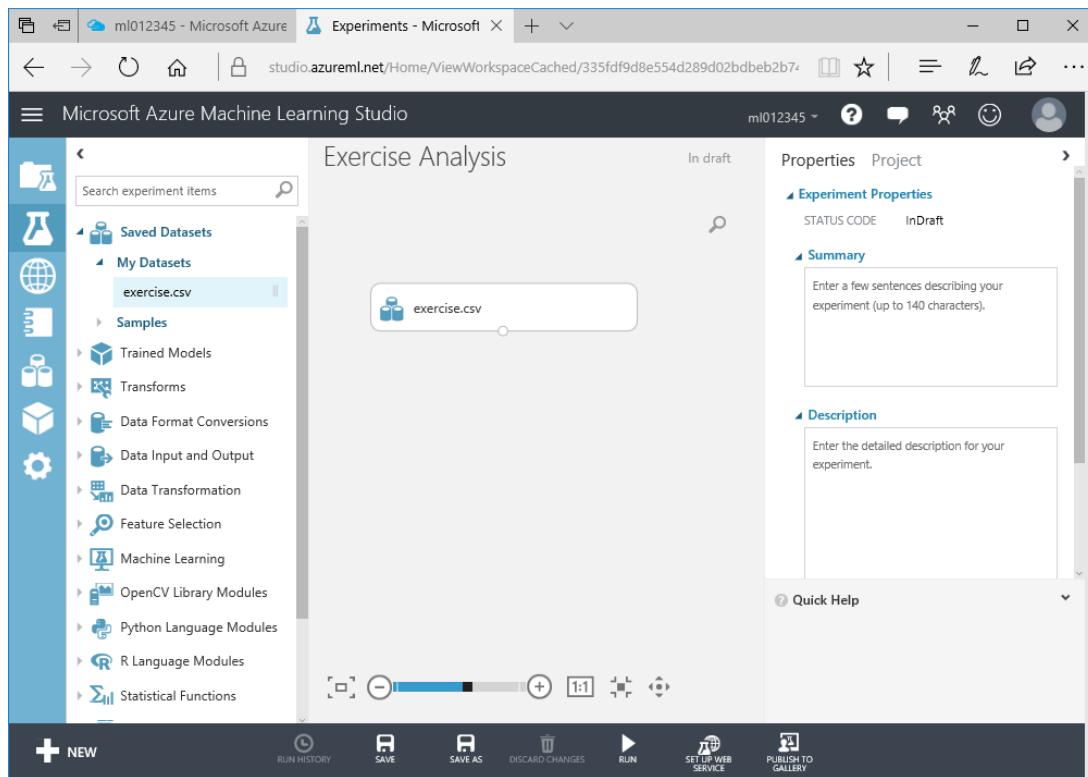
The exercise data is provided as a comma-delimited text file.

1. In Azure Machine Learning Studio, click **DATASETS**. You should have no datasets of your own (clicking **Samples** will display some built-in sample datasets).
2. At the bottom left, click **+ NEW**, and ensure that the **DATASET** tab is selected.
3. Click **FROM LOCAL FILE**. Then in the **Upload a new dataset** dialog box, browse to select the **exercise.csv** file in the **Lab01** folder where you extracted the lab files on your local computer and enter the following details as shown in the image below, and then click the (✓) icon.
 - **This is a new version of an existing dataset:** Unselected
 - **Enter a name for the new dataset:** exercise.csv
 - **Select a type for the new dataset:** Generic CSV file with a header (.csv)
 - **Provide an optional description:** Exercise data.
4. Wait for the upload of the dataset to be completed, then verify that it is listed.

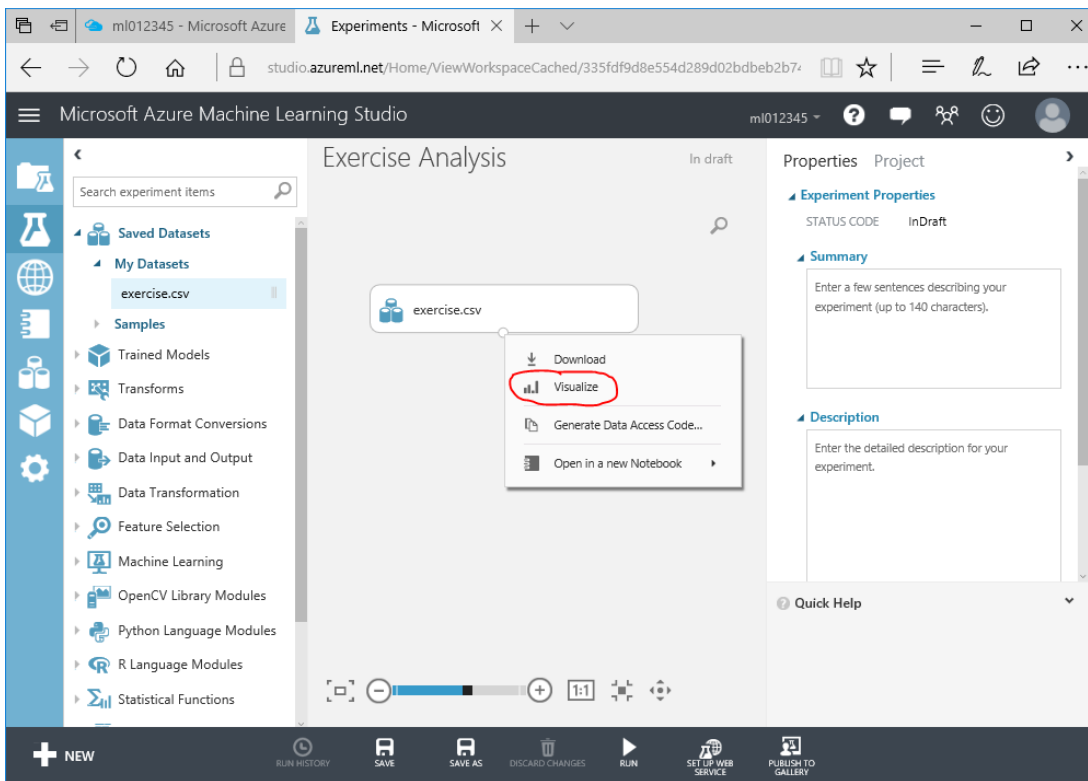
Create an Experiment

Now that you have uploaded your data, you can create an experiment to explore it.

1. In Azure Machine Learning Studio, click **EXPERIMENTS**. You should have no experiments in your workspace yet.
2. At the bottom left, click **+ NEW**, and ensure that the **EXPERIMENT** tab is selected. Then click the **Blank Experiment** tile to create a new blank experiment.
3. At the top of the experiment canvas, change the experiment name to **Exercise Analysis**.
4. In the experiment items pane, expand **Saved Datasets** and **My Datasets**, and then drag the **exercise.csv** dataset onto the experiment canvas, as shown here:

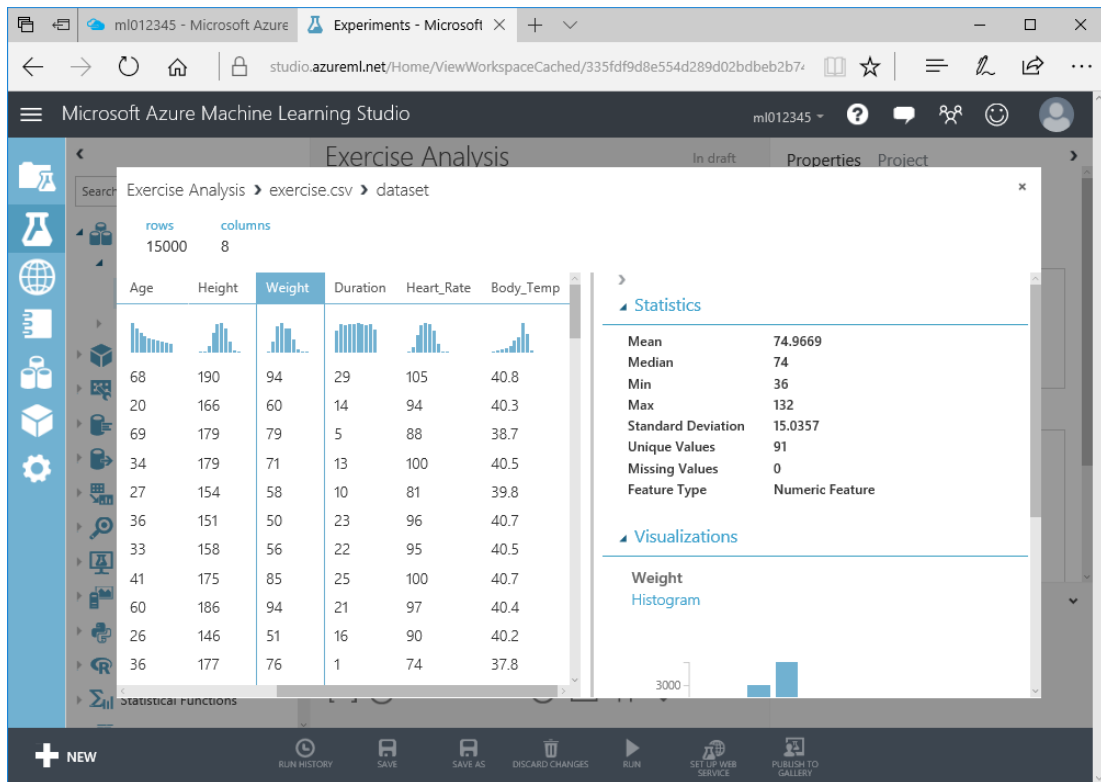


5. Right-click the dataset output of the **exercise.csv** dataset and click **Visualize** as shown here:

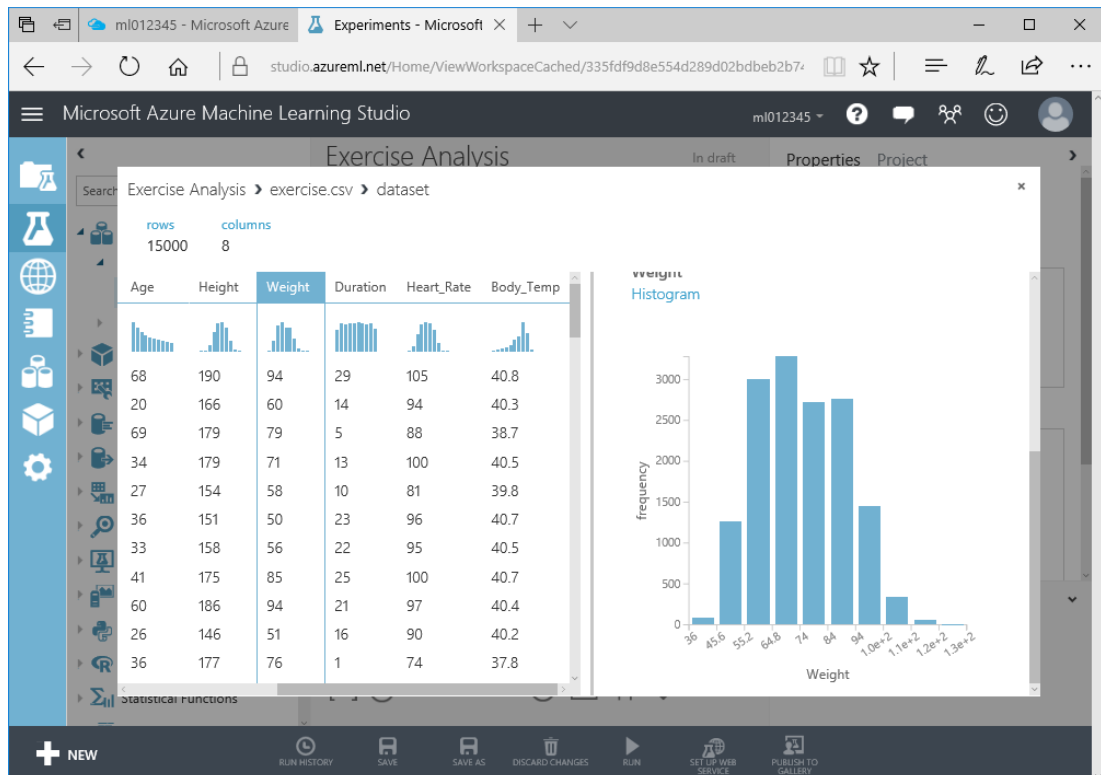


6. In the data visualization, note that the dataset includes a record for each participant (identified by a User ID), and each record includes physiological measurements for the user during a period of exercise. Note the number of rows and columns in the dataset, and then select the column

heading for the **Weight** column and note the statistics about that column that are displayed, as shown here:



7. In the data visualization, scroll down if necessary to see the histogram for weight. This shows the distribution of different weights within the patient records in the dataset.



Note: This distribution is close to what data scientists call a *normal* distribution, in which the most frequently occurring values for a variable (in this case weight) tend to be in the middle of the range, with approximately similar rates of drop-off as the values move towards the extreme high and low ends.

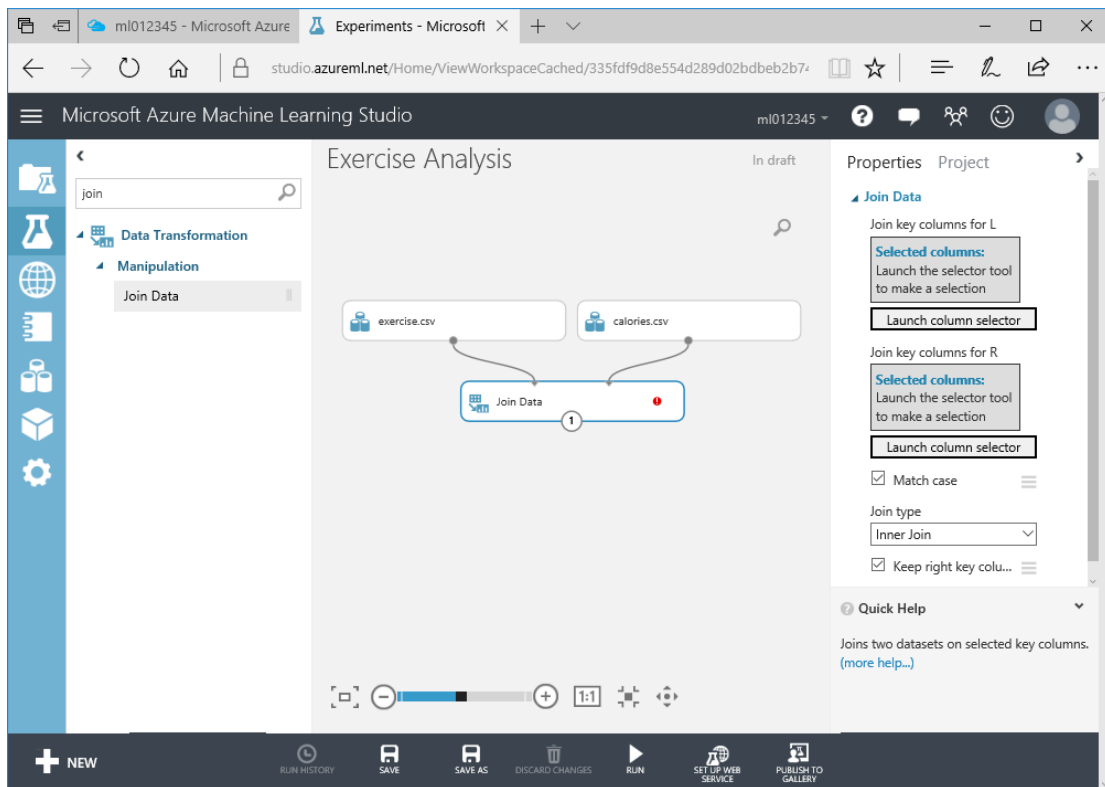
8. Close the visualization and return to the experiment canvas.

The actual calories expended by the study participants is stored in a separate file, which you must upload as a dataset and add to the experiment.

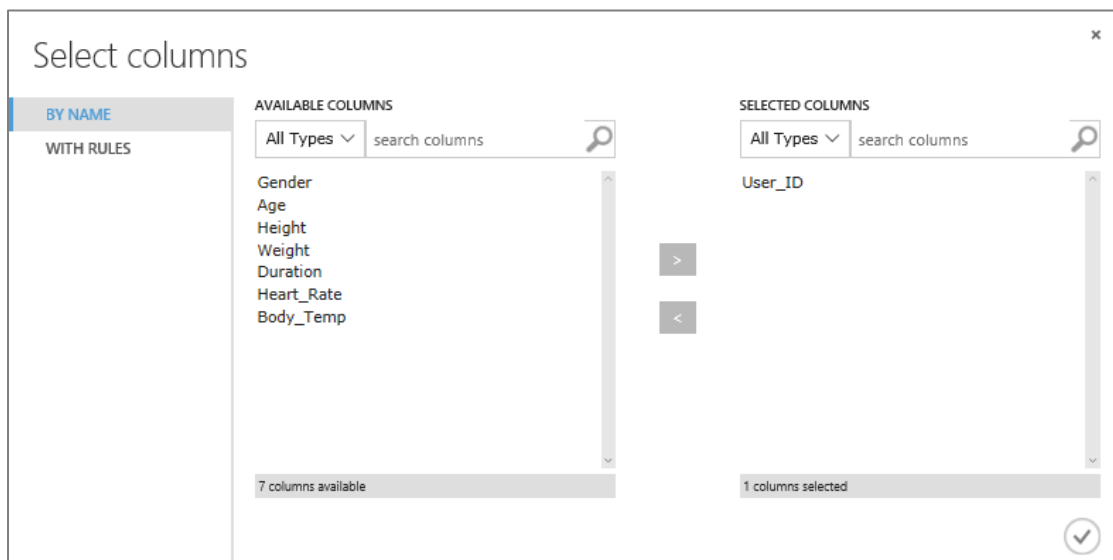
5. At the bottom left, click **+ NEW**, and select the **DATASET** tab.
6. Click **FROM LOCAL FILE**. Then in the **Upload a new dataset** dialog box, browse to select the **calories.csv** file in the **Lab01** folder where you extracted the lab files on your local computer and enter the following details as shown in the image below, and then click the (✓) icon.
 - **This is a new version of an existing dataset:** Unselected
 - **Enter a name for the new dataset:** calories.csv
 - **Select a type for the new dataset:** Generic CSV file with a header (.csv)
 - **Provide an optional description:** Calorie data.
9. Wait for the upload of the **calories.csv** dataset to be completed, then verify that it is listed in the **My Datasets** node.
10. Drag the **calories.csv** dataset to the experiment canvas, to the right of the **exercise.csv** dataset.
11. Visualize the output of the calories.csv and note that it contains observations for study participants and the calories they expended during the exercise periods. View the statistics and histogram for the **Calories** column.
12. Close the visualization and return to the experiment canvas.

Now your experiment contains two datasets with a common **User_ID** field. You can use this field to combine the two datasets.

13. In the **Search experiment items** box, type *Join*, and then from the filtered items list, drag the **Join Data** module onto the canvas and place it below the two datasets.
14. Connect the dataset output from the **exercise.csv** dataset to the **Dataset1** (left) input of the **Join Data** module, and connect the dataset output from the **calories.csv** dataset to its **Dataset2** (right) input as shown here:

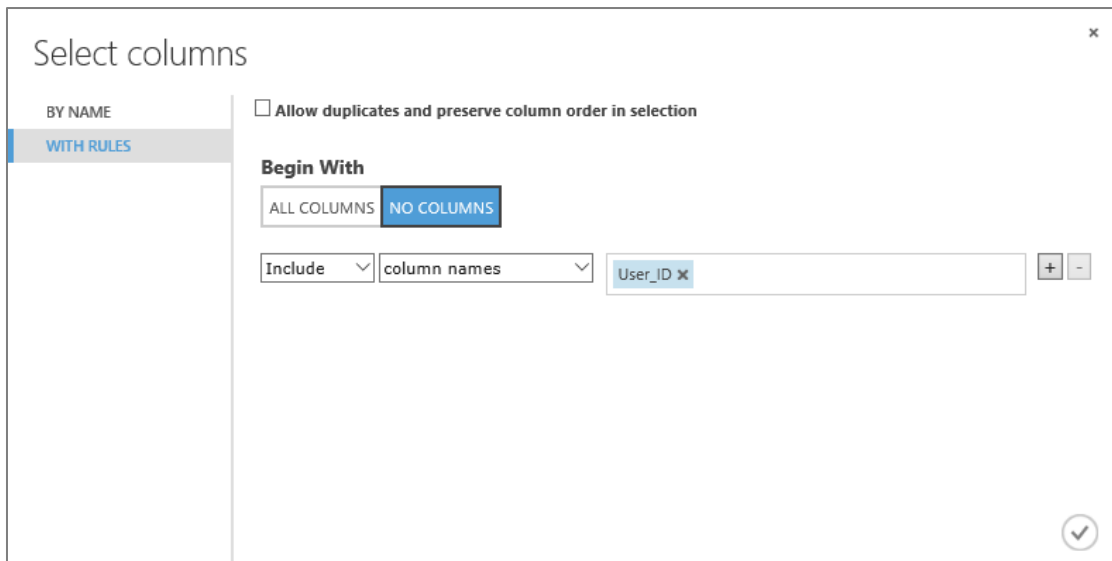


15. With the **Join Data** module selected, in the **Properties** pane, under **Join key columns for L**, click **Launch column selector**.
16. With the **BY NAME** tab selected, select the **User_ID** column and click [**>**] to add it to the **Selected Columns** list, as shown here:



17. Click the (✓) icon to confirm the column selection.
18. With the **Join Data** module selected, in the **Properties** pane, under **Join key columns for R**, click **Launch column selector**.
19. Click the **WITH RULES** tab, and note that you can define rules to select columns based on their name, index, and data type. Under **Begin With**, select **No Columns**. Then select **Include, column**

names, and **User_ID** as shown here (the column names should appear when you start to type). Then click the (✓) icon.



Select columns

BY NAME
WITH RULES

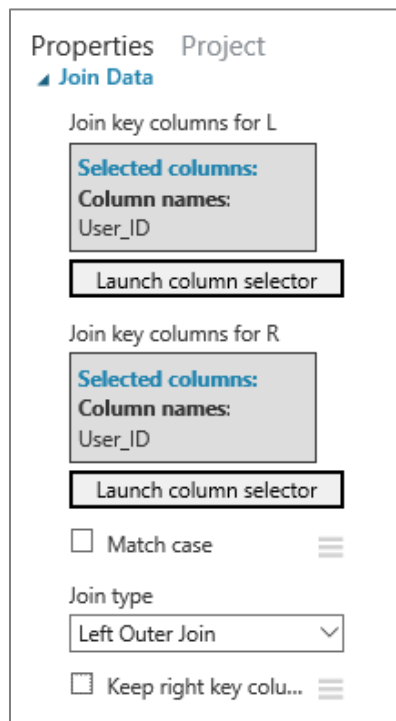
☐ Allow duplicates and preserve column order in selection

Begin With
ALL COLUMNS NO COLUMNS

Include column names User_ID x

✓

20. With the **Join Data** module selected, in the **Properties** pane, clear the **Match case** checkbox, select **Left Outer Join**, and clear the **Keep right key column in joined table** checkbox as shown here:



Properties Project

Join Data

Join key columns for L

Selected columns:
Column names:
User_ID

Launch column selector

Join key columns for R

Selected columns:
Column names:
User_ID

Launch column selector

☐ Match case

Join type
Left Outer Join

☐ Keep right key column in joined table

Note: Using a left outer join ensures that the joined table includes all users in the **exercise.csv** dataset and their corresponding calories measurement from the **calories.csv** dataset. If there are any observations in the **exercise.csv** dataset with no matching **calories.csv** record, the exercise data will be retained, and the corresponding calories value will be NULL.

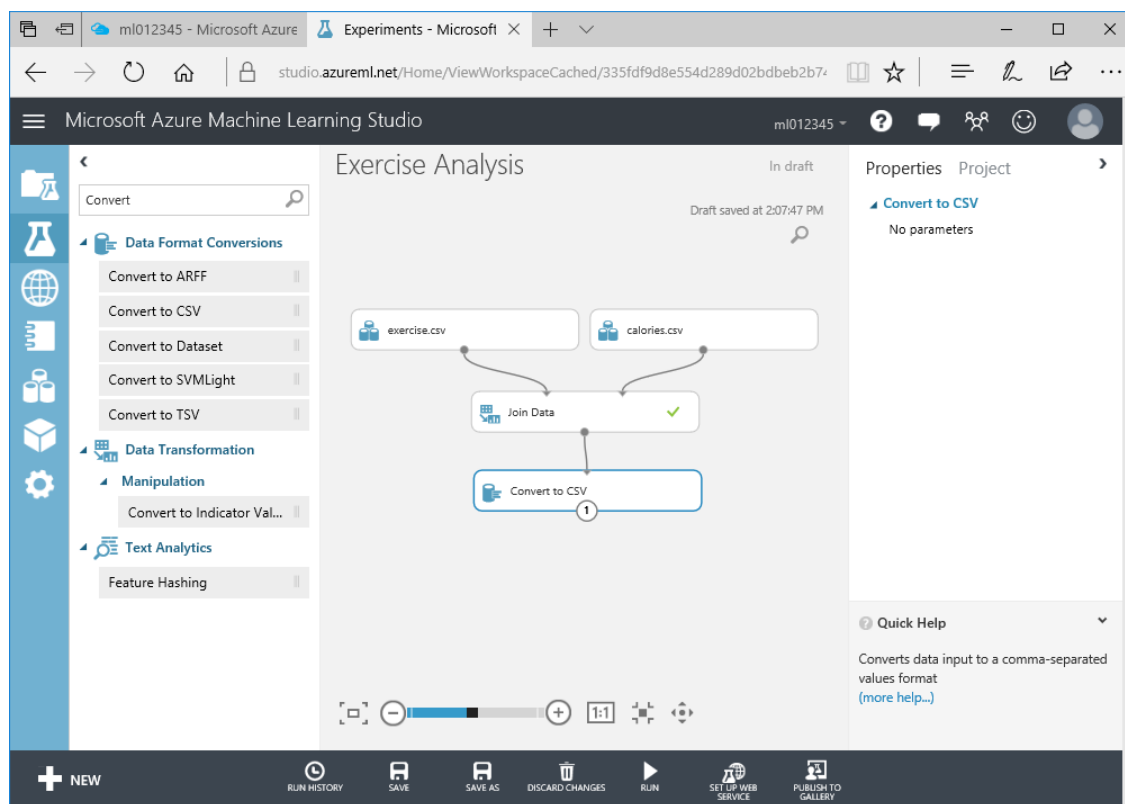
21. On the toolbar at the bottom of the experiment canvas, click **Save**. Then click **Run** to run the experiment. Wait for the experiment to finish running (a green ✓ icon will be displayed in the **Join Data** module)
22. Visualize the **Results dataset** output from the **Join Data** module, and note that it contains all the **exercise.csv** columns and the corresponding **Calories** column from the **calories.csv** dataset.
23. Close the visualization.

Note: Azure Machine Learning experiments typically include multiple modules that form a data flow in which data from a dataset is cleaned, filtered, and otherwise prepared for analysis or modeling. Azure Machine Learning includes a wide range of modules for common data operations as well as modules that enable you to implement custom logic in Python, R, or SQL.

Create a Notebook

Notebooks provide a convenient way for data scientists to explore data using R or Python code.

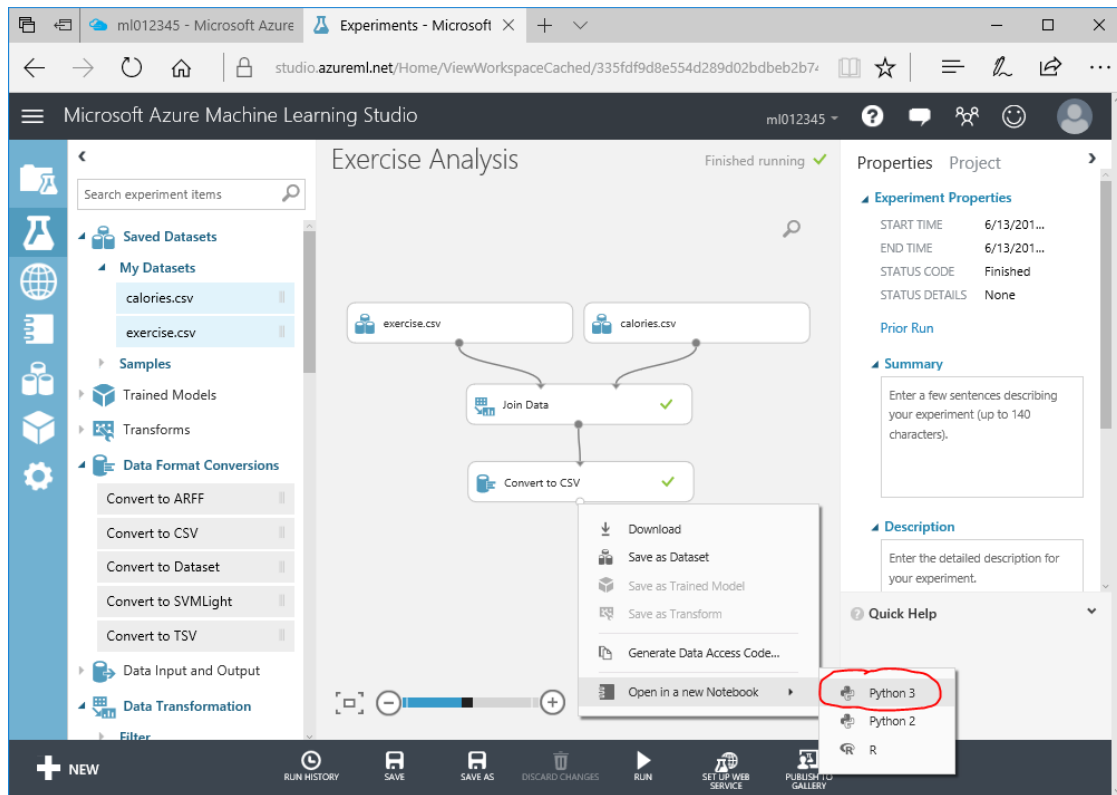
1. In the **Search experiment items** box, type *Convert*, and then from the filtered items list, drag the **Convert to CSV** module onto the canvas and place it below the **Join Data** module.
2. Connect the **Results dataset** output from the **Join Data** module to the **Dataset** input of the **Convert to CSV** module as shown here:



Note: You can open a CSV dataset directly in a notebook. If you have applied any transformations to the data, you must convert it back to CSV before opening it in a notebook.

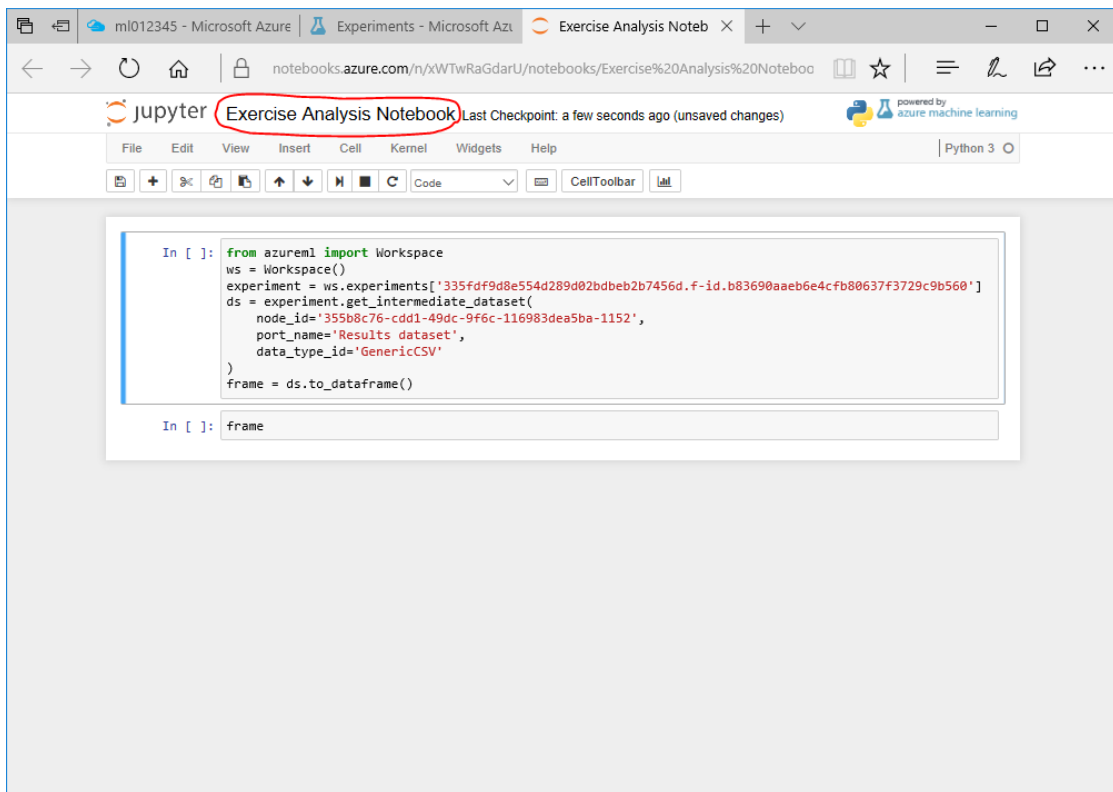
3. Select the **Convert to CSV** module, then on the **Run** menu, click **Run Selected**. This runs the selected module using the upstream output from the previous run. Wait until it has finished running.

- Right-click the **Result dataset** output of the **Convert to CSV** module, and then in the **Open in a new Notebook** menu, click **Python 3**; as shown here:



Note: Azure Machine Learning Jupyter notebooks currently support custom code written in R, Python 2, and Python 3. Data scientists can use whichever language they prefer.

- In the new browser tab that opens, view the Jupyter notebook that has been created. Note that by default, the notebook has a title similar to **Run result 5-30-2017 3_33_13 PM Python 3 notebook**. Click this title and rename the notebook to **Exercise Analysis Notebook**, as shown here:



6. Observe that the notebook contains two cells. The first cell contains code that loads the CSV dataset into a data frame named **frame**, similar to this:

```
from azureml import Workspace
ws = Workspace()
experiment = ws.experiments['335fdf9...f-id.0d127ca2e02245...']
ds = experiment.get_intermediate_dataset(
    node_id='ee772e8e-1600-49a5-a615-a6cfd9661208-21',
    port_name='Results dataset',
    data_type_id='GenericCSV'
)
frame = ds.to_dataframe()
```

The second cell contains the following code, which displays a summary of the data frame:

```
frame
```

7. On the **Cell** menu, click **Run All** to run all of the cells in the workbook. As the code runs, the **Python 3** symbol next to **Python 3** at the top right of the page changes to a **●** symbol, and then returns to **○** when the code has finished running.
8. Observe the output from the second cell, which shows the first few rows of data from the dataset, as shown here:

```

In [1]: from azureml import Workspace
ws = Workspace()
experiment = ws.experiments['335fd9d8e554d289d02bdb2b7456d.f-id.b83690aaeb6e4cfb80637f3729c9b560']
ds = experiment.get_intermediate_dataset(
    node_id='355b8c76-cdd1-49dc-9f6c-116983dea5ba-1152',
    port_name='Results dataset',
    data_type_id='GenericCSV'
)
frame = ds.to_dataframe()

In [2]: frame
Out[2]:

```

	User_ID	Gender	Age	Height	Weight	Duration	Heart_Rate	Body_Temp	Calories
0	14733363	male	68	190	94	29	105	40.8	231
1	14861698	female	20	166	60	14	94	40.3	66
2	11179863	male	69	179	79	5	88	38.7	26
3	16180408	female	34	179	71	13	100	40.5	71
4	17771927	female	27	154	58	10	81	39.8	35
5	15130815	female	36	151	50	23	96	40.7	123
6	19602372	female	33	158	56	22	95	40.5	112
7	11117088	male	41	175	85	25	100	40.7	143
8	12132339	male	60	186	94	21	97	40.4	134
9	17964668	female	26	146	51	16	90	40.2	72

9. Click cell 2 (which contains the code `frame`), and then on the **Insert** menu, click **Insert Cell Below**. This adds a new cell to the notebook, under the output generated by cell 2.
10. Add the following code to the new empty cell (you can copy and paste this code from **NotebookCode.txt** in the **Lab01** folder):

```

# Create a scatter plot matrix
%matplotlib inline

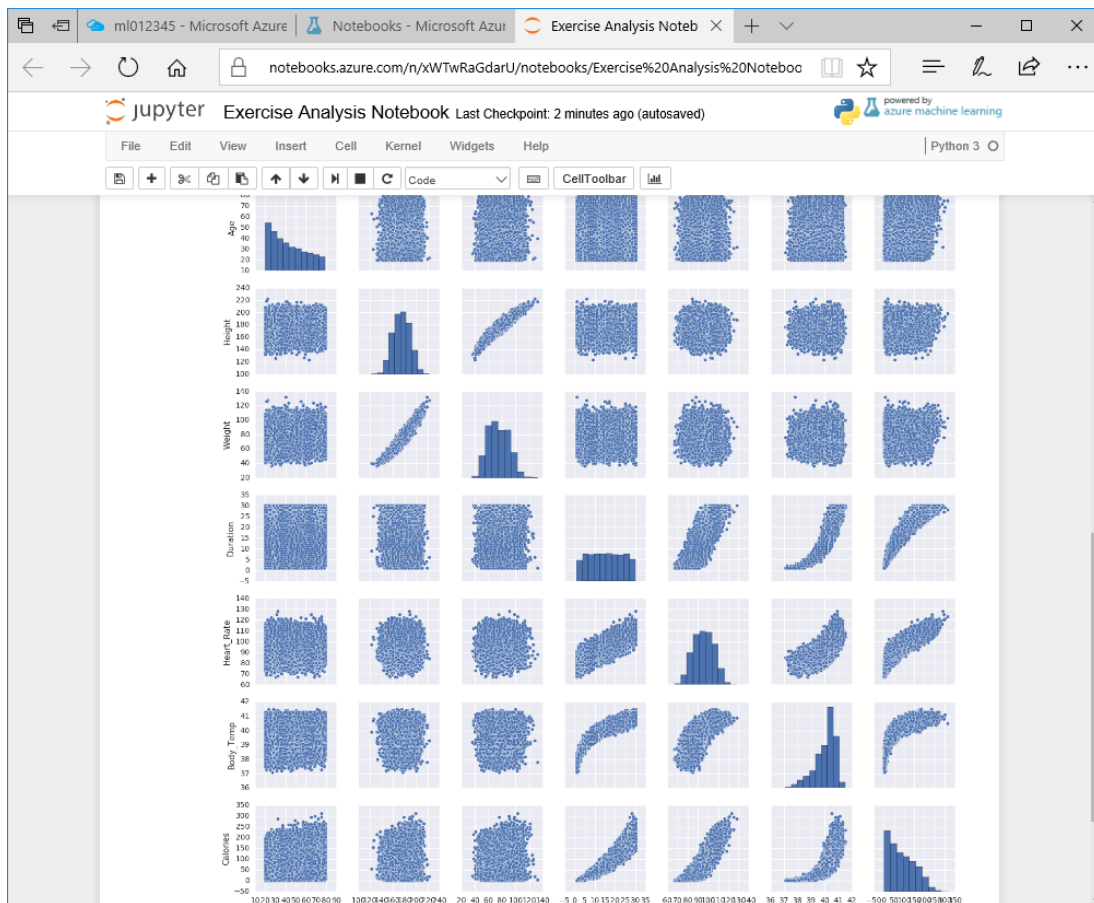
import seaborn as sns
num_cols = ["Age", "Height", "Weight", "Duration",
            "Heart_Rate", "Body_Temp", "Calories"]
sns.pairplot(frame[num_cols], size=2)

```

11. With the cell containing the new code selected, on the **Cell** menu, click **Run Cells and Select Below** (or click the **▶ |** button on the toolbar) to run the cell, creating a new cell beneath.

Note: You can ignore the warnings that are generated.

12. View the output from the code, which consists of a scatter plot matrix, as shown here:

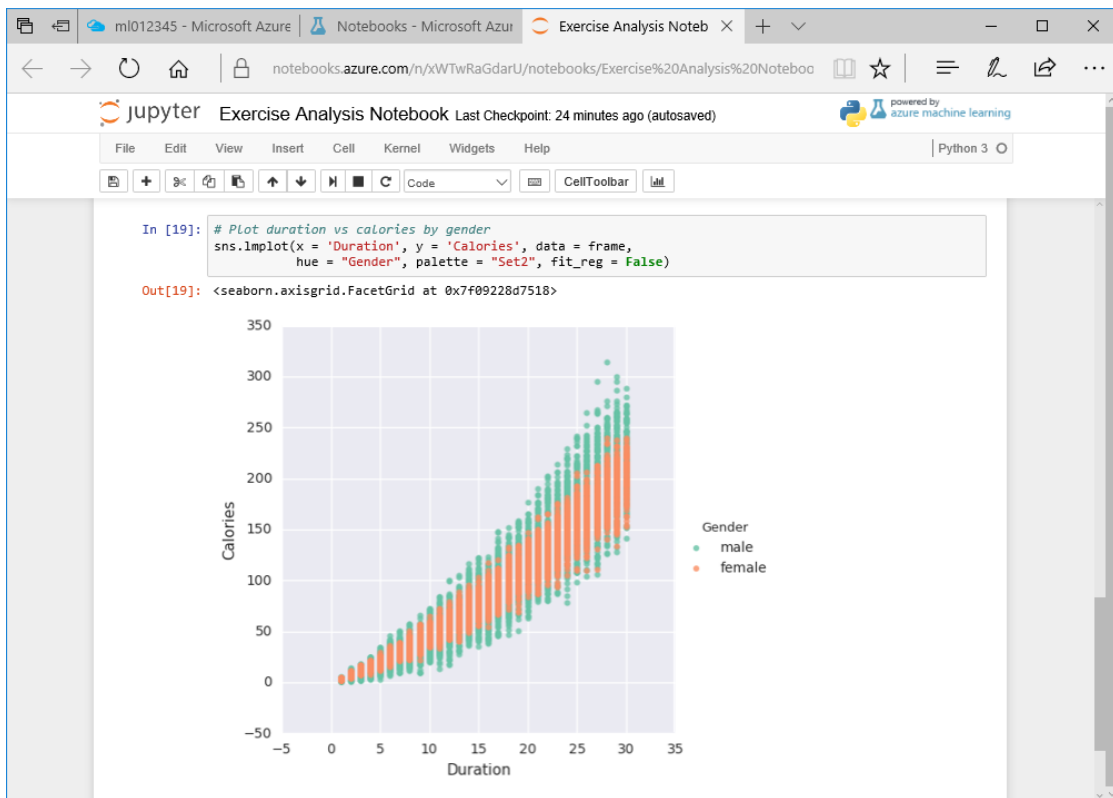


Note that from this diagram, you can clearly see some apparent relationships in the data. For example, as height increases, so does weight; as exercise duration increases, so does body temperature; and as heart rate increases, so does the number of calories expended.

13. In the empty cell at the end of the notebook, add the following code (you can copy and paste this code from **NotebookCode.txt** in the **Lab01** folder):

```
# Plot duration vs calories by gender
sns.lmplot(x = 'Duration', y = 'Calories', data = frame,
           hue = "Gender", palette = "Set2", fit_reg = False)
```

14. Run the cell, and view the output from the code, which consists of a scatter plot of **Duration** vs **Calories**, conditioned by **Gender**, as shown here:



Note that from this plot, you can verify that the number of calories expended seems to increase in line with the duration of the exercise period for both female and male participants. However, the number of calories burned by males tends to vary across a wider range than those burned by females.

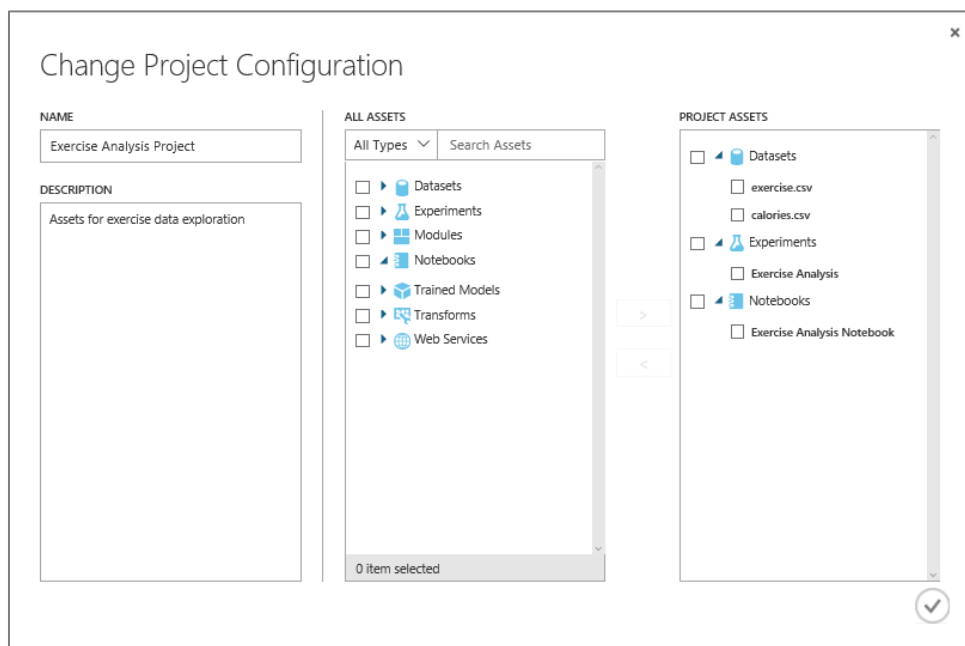
15. On the **File** menu, click **Save and Checkpoint** to save the notebook. Then on the **File** menu, click **Close and Halt** to close the notebook.
16. In Azure Machine Learning Studio, click the **Notebooks** page on the left and verify that the notebook is listed there.

Note: The notebook name may still be listed as **Run result....** If this is the case, select it and click **Rename** at the bottom of the page to rename it to **Exercise Analysis Notebook**.

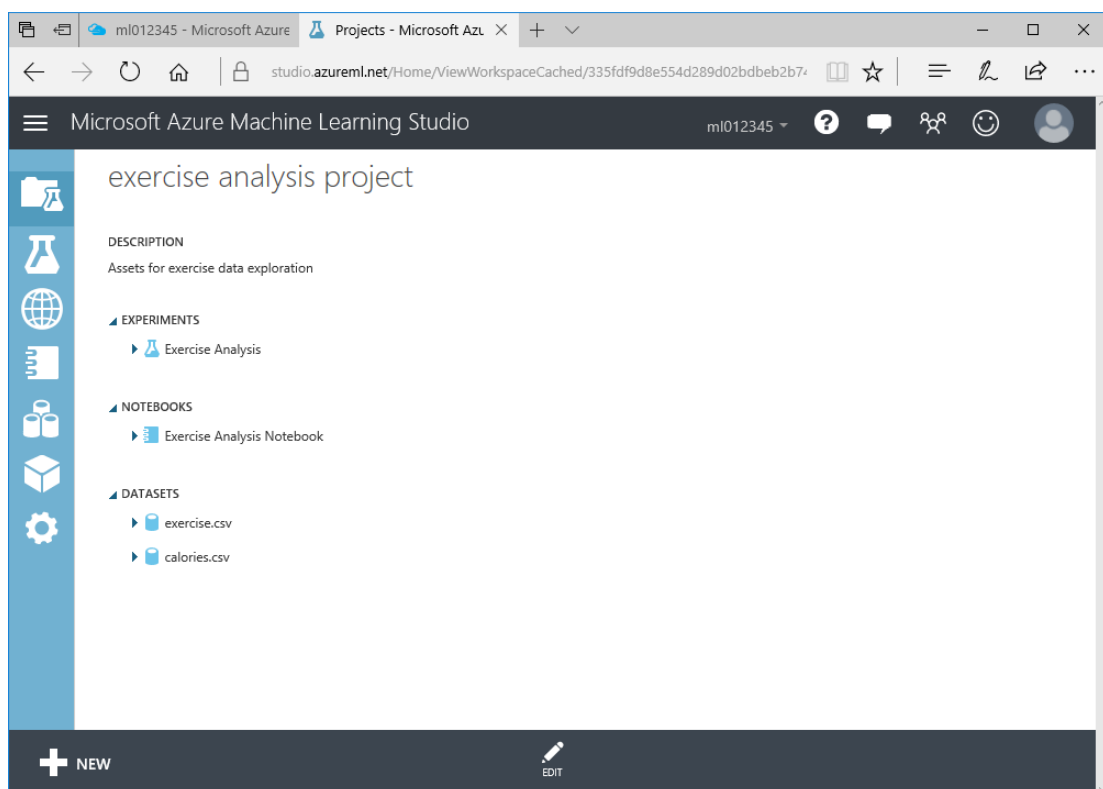
Create a Project

Projects are a useful way to group related datasets, experiments, notebooks, and other assets.

1. In Azure Machine Learning Studio, click the **Projects** page. Then click **Create project**.
2. Name the new project **Exercise Analysis Project**, and add the description **Assets for exercise data exploration**.
3. In the **Exercise Analysis Project** page, click **Add assets**.
4. In the list of all assets, expand **Datasets** and select **exercise.csv** and **calories.csv**; expand **Experiments** and select **Exercise Analysis**, and expand **Notebooks** and select **Exercise Analysis Notebook**. Then click **>** to add these assets to the project as shown here:



- Click (✓) to save the changes, and verify that **Exercise Analysis Project** contains the experiment, notebook, and dataset you created in this exercise as shown here:



Exercise 3: Working with Big Data Sources

In a big data solution, large volumes of data are typically cleaned and prepared for analysis by using batch processes that leverage parallelism and operate on the data in big data stores. After the data has been prepared by the big data processing operations, you can ingest it into an Azure Machine Learning

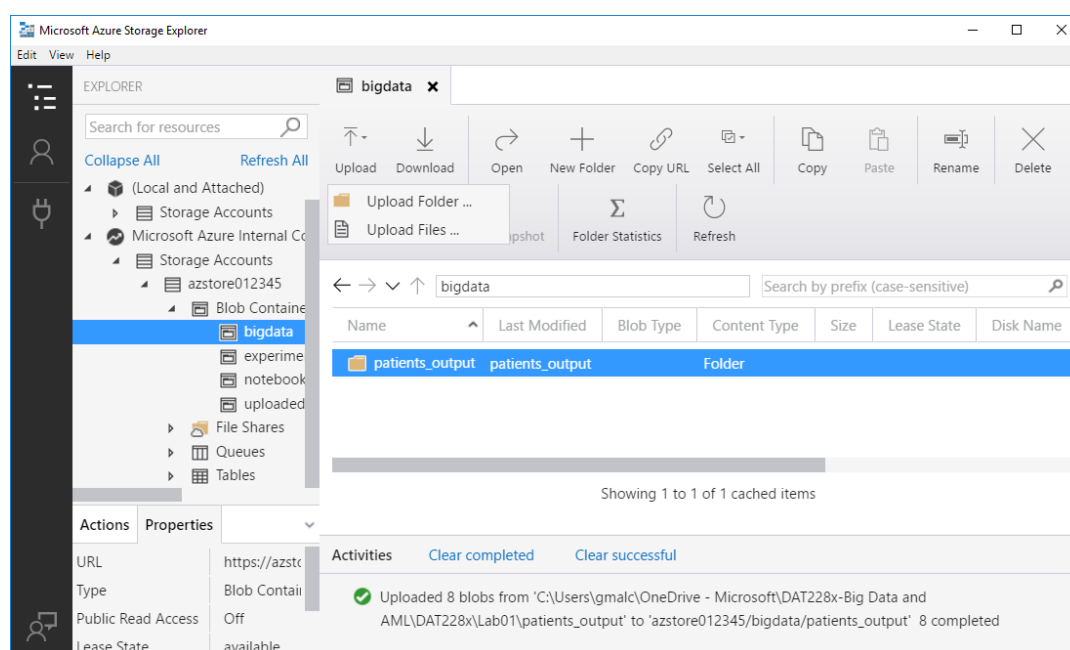
experiment. In this exercise, you will create an Azure Machine Learning experiment that ingests data from external data sources typically found in an Azure-based big data architecture.

Note: The data used in the exercise was generated by a simulation based on the data in the *Pima Indians Diabetes* dataset published by the University of California, School of Information and Computer Science at <https://archive.ics.uci.edu/ml/datasets/Pima+Indians+Diabetes>.

Upload Data to Azure Storage

Azure storage is a commonly used data store for cloud applications, and is often the output destination for big data processing; including Hadoop or Spark jobs run in Azure HDInsight, or U-SQL jobs run in Azure Data Lake Analytics. For example, a clinic conducting diabetes research might gather patient data during the day and then run a batch processing job each night to clean and prepare the data for analysis, producing text files in an Azure storage blob container. In this procedure, you will upload text files that represent the output of a big data job to Azure storage.

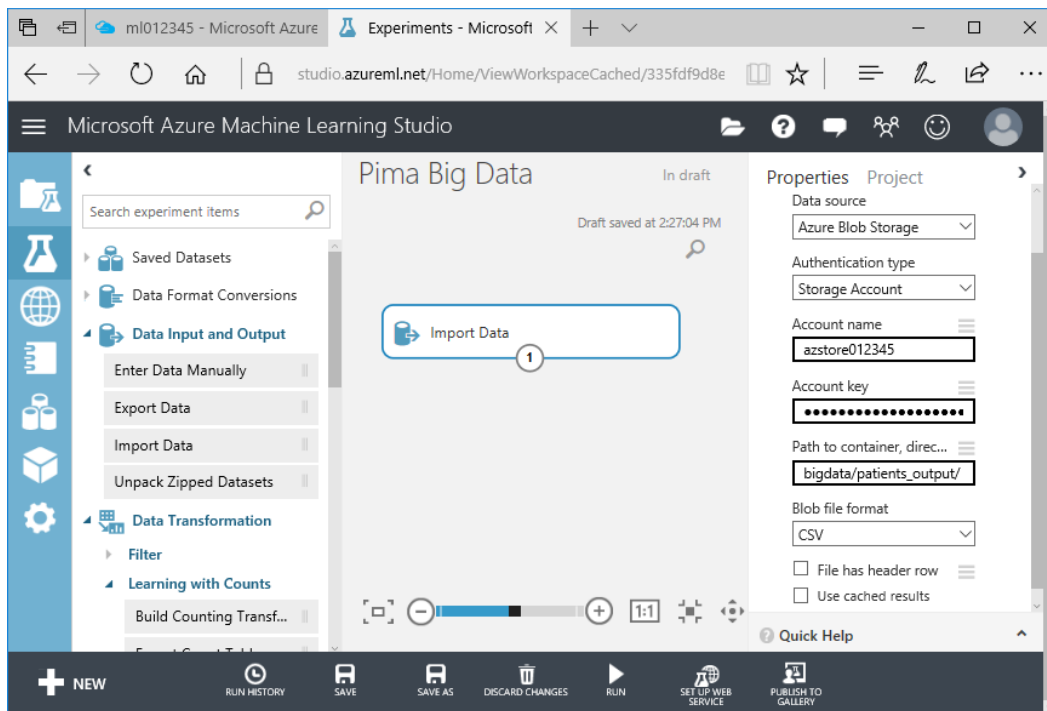
1. In the **Lab01** folder where you extracted the lab files for this course, view the contents of the **patient_output** folder. This folder contains multiple text files, similar to the output that would be generated by a big data processing job to prepare the diabetes patient data.
2. Use a text editor to view on the files, and note that it contains data but no column headings.
3. Start Azure Storage Explorer, and if you are not already signed in, sign into your Azure subscription.
4. Expand your storage account, right-click the **Blob Containers** folder and click **Create Blob Container**. Name your new blob container **bigdata**.
5. With the new **bigdata** blob container selected, in the **Upload** drop-down list, click **Upload Folder**. Then upload the **patients_output** folder to a new folder named **patients_output**, as shown here:



Import Data from Azure Storage into an Experiment

Now that you have some data in Azure storage that represents the output from a big data processing job, you can use it in an Azure Machine Learning experiment.

1. In Azure Machine Learning Studio, on the **Experiments** page, create a new blank experiment.
2. Name the new experiment **Pima Big Data**.
3. Add an **Import Data** module to the experiment canvas.
4. With the **Import data** module selected, in the **Properties** pane, click **Launch Import Data Wizard**.
5. In the **Choose data source** page, select **Azure Blob Storage** and click the (→) icon.
6. In the **Connect to Azure Blob Storage** page, set the following properties and click the (→) icon.
 - **Authentication type:** Storage Account
 - **Subscription ID:** The subscription containing your storage account.
 - **Account name:** Your Azure Storage account.
 - **Account key:** Select the *Primary Key* for your storage account.
 - **Path to container, directory, or blob:** bigdata/patients_output/
7. In the **Configure import from Azure Blob** page, in the **Blob file format** list, select **CSV**. Then ensure that the **File has header row** checkbox is not selected, and click the (✓) icon.
8. View the **Properties** pane for the **Import Data** module, which should resemble the following:

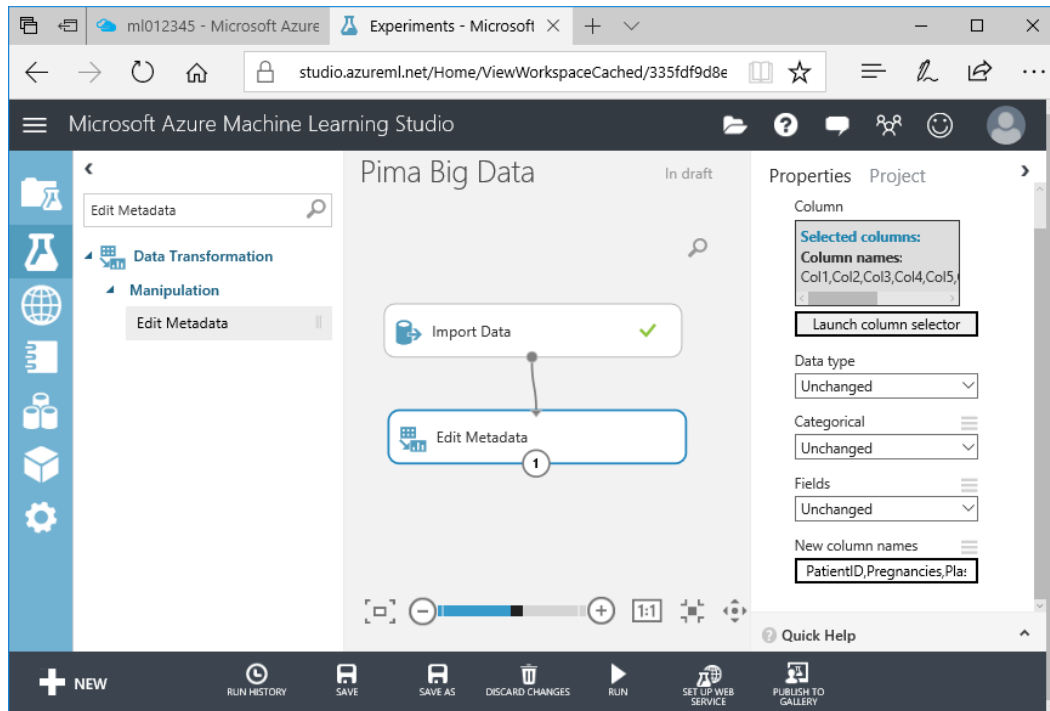


Note: You can find the keys for an Azure Storage account in its blade in the Azure portal.

9. Save and run the experiment.
10. When the experiment has finished running, visualize the **Results dataset** output from the **Import Data** module. Note that the columns in the dataset have been assigned the names **Col1**, **Col2**, **Col3**, and so on.
11. Add an **Edit Metadata** module to the experiment, and connect the output from the **Import Data** module to its input.
12. Edit the properties of the **Edit Metadata** module to select all columns, and in the **New Column names** box, enter the following comma-delimited list of column names:

PatientID,Pregnancies,PlasmaGlucose,DiastolicBloodPressure,TricepsThickness,SerumInsulin,BMI,DiabetesPedigree,Age,Diabetic

13. Verify that the experiment looks like this, and then save and run the experiment:



14. When the experiment has finished running, visualize the **Results dataset** output from the **Edit Metadata** module. Note that the columns have been renamed based on the values you entered.

Provision Azure SQL Database

Many big data solutions store data in a relational database, such as Azure SQL Database or Azure SQL Data Warehouse. In this procedure, you will provision an Azure SQL Database to store details of the physicians associated with the patients in the diabetes research project.

1. In the browser pane containing the Microsoft Azure portal, in the menu, click **New**. Then in the **Databases** menu, click **SQL Database**.
2. In the **SQL Database** blade, enter the following settings, and then click **Create**:
 - **Database name**: DiabetesData
 - **Subscription**: *Select your Azure subscription*
 - **Resource Group**: *Select the resource group you created previously*
 - **Select source**: Blank database
 - **Server**: *Create a new server with the following settings:*
 - **Server name**: *Enter a unique name (and make a note of it!)*
 - **Server admin login**: *Enter a user name of your choice (and make a note of it!)*
 - **Password**: *Enter and confirm a strong password (and make a note of it!)*
 - **Region**: *Select the same location as your storage account*
 - **Allow azure services to access server**: Selected
 - **Elastic pool**: *Not enabled*
 - **Pricing tier**: *View all and select Basic*
 - **Collation**: SQL_Latin1_General_CP1_CI_AS
 - **Pin to dashboard**: Unselected
3. In the Azure portal, view **Notifications** to verify that deployment has started. Then wait for the SQL database to be deployed (this can take a few minutes.)

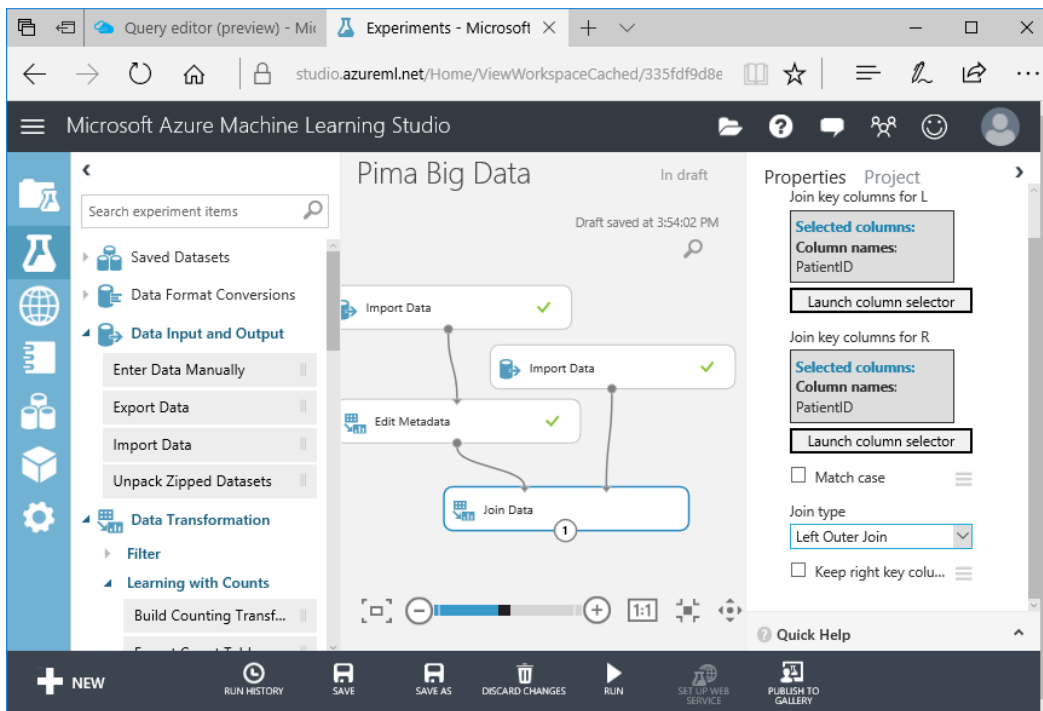
4. After the database has been created, browse to the blade for the **DiabetesData** database, and then on the **Tools** menu, click **Query editor** to open the web-based query interface for your database.
5. In the **Query editor** blade, click **Login** and then log in to your Azure SQL database server using **SQL server authentication** with the username and password you specified when provisioning the database.
6. After you have been authenticated, copy and paste the code in the **CreateDoctorTable.txt** script file in the **Lab01** folder into the empty query editor. This script creates a table named **dbo.Doctors** and inserts some data into it.
7. Click **Run**, and then wait for the query to complete. This may take a while.
8. When the query has completed, close the query editor blade, discarding the changes to your script.

Import Data from Azure SQL Database into an Experiment

Now that you have loaded some data into a database table, you can ingest it into an Azure Machine Learning experiment.

1. Return to the **Pima Big Data** experiment in Azure Machine Learning Studio.
2. Add a second **Import Data** module to the experiment, and use the **Import Data Wizard** to configure it as follows:
 - **Choose Data Source:** Azure SQL database
 - **Connect to Azure SQL Database:**
 - **Subscription ID:** The subscription containing your database.
 - **Database server name:** The server you created for your database.
 - **Database name:** DiabetesData
 - **User name:** The user name you specified for your login.
 - **Password:** The password you specified for your login
 - **Database query:**

```
SELECT PatientID, Physician
FROM dbo.Doctors;
```
3. With the new **Import Data** module selected, in the **Run** menu, click **Run Selected**. Then when the experiment has finished running, visualize the output of the new **Import Data** module to view the physician data.
4. Add a **Join Data** module to the experiment, and connect the output from the **Edit Metadata** module containing the patient data to its left input, and the output from the **Import Data** module containing the physician data to its right input.
5. Configure the properties of the **Join Data** module as follows:
 - **Join key columns for L:** PatientID
 - **Join key columns for R:** PatientID
 - **Match case:** Unselected
 - **Join type:** Left Outer Join
 - **Keep right key column in joined table:** Unselected
6. Verify that your experiment looks like this, and then save and run it:



7. Visualize the output from the **Join Data** module and verify that each patient record now includes the name of their physician.

Note: By using a right outer join, the experiment will retain patients for whom no physician is assigned, entering NULL in the **Physician** column for such patients.

Summary

In this lab, you have explored Azure Machine Learning Studio, uploaded datasets, and created experiments. You have also used various modules to import data from big data sources and define a data workflow in an experiment.

Now you're ready to learn more about machine learning, and training predictive models in Azure Machine Learning. In the next lab of this course, you will build on the work you have done in this lab.