

Create an Experiment in Azure Machine Learning Studio

Introduction

Microsoft Azure Machine Learning Studio is a drag-and-drop interface that allows you to create experiments for AI application consumption as web services. Machine Learning Studio provides many built-in statistics and functions for machine learning experiments.

Learning objectives

In this module you will:

- Create an experiment.
- Prepare and publish a web service for the experiment.
- Test the web service.
- View the published web service headers and sample code.

Prerequisites

If you don't have a Machine Learning Studio account, complete the Microsoft Learn module [Create an Azure Machine Learning Studio account](#) before you begin.

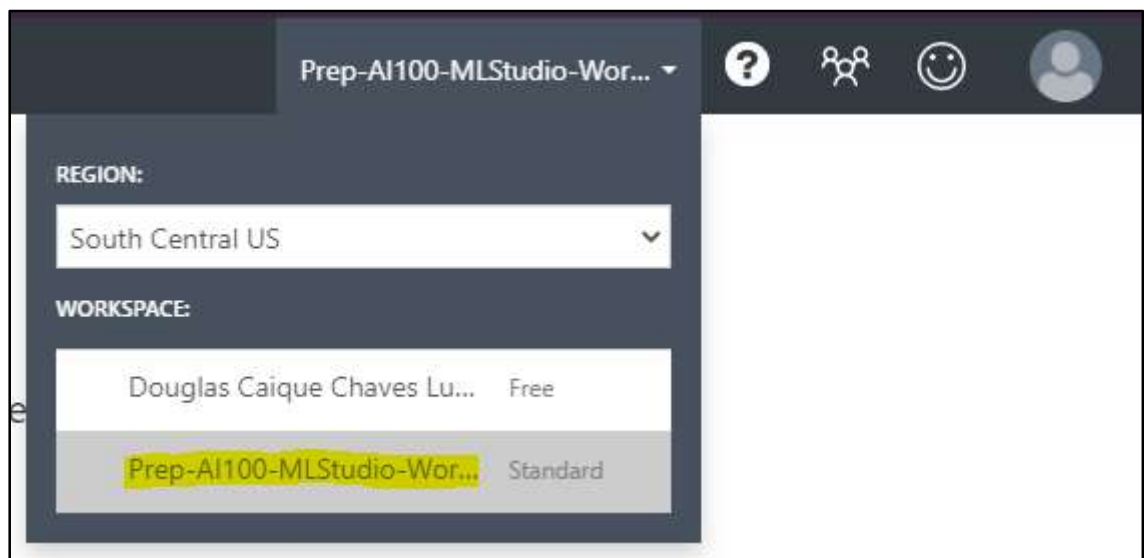
Exercise - Create an experiment

We'll start in Azure Machine Learning Studio by creating an experiment. The experiment uses a prebuilt dataset that relates to building energy considerations.

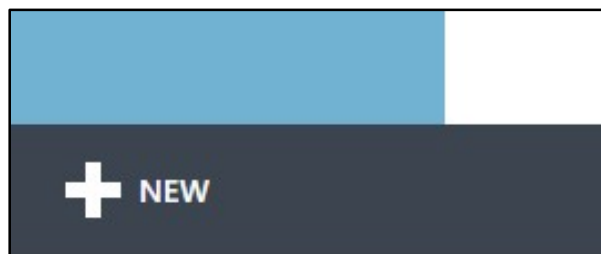
In this exercise, you'll import a dataset into Machine Learning Studio. There you'll experiment with it to publish a web service. This simple experiment shows how to create a machine learning model that you can publish and consume as a web service.

To import a dataset into Machine Learning Studio:

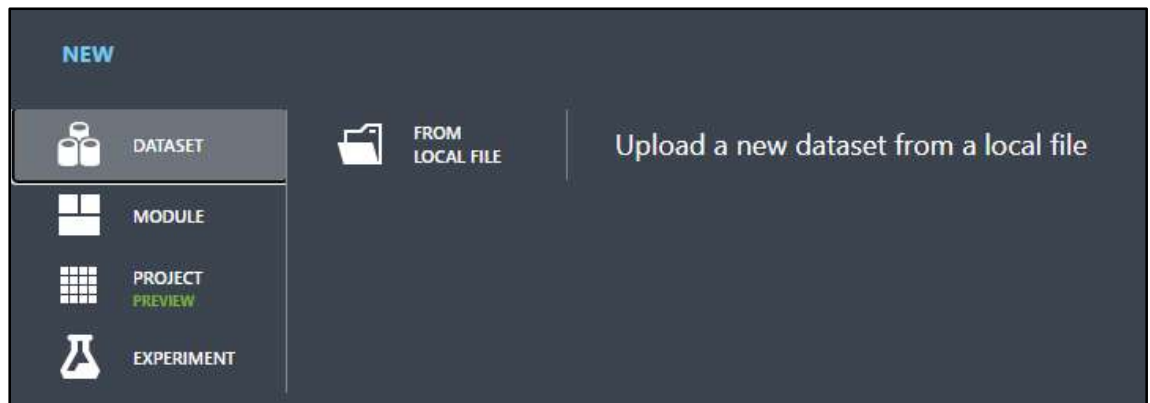
1. Download and extract the [Building Data.zip](#) file. This is the dataset.
2. Go to the [Machine Learning Studio portal](#).



3. Sign in by using the Microsoft account that's associated with your Azure account.
4. On the left navigation bar of Machine Learning Studio, select the **Datasets** icon.
5. In the lower-left corner, select the **New** button to upload the dataset.



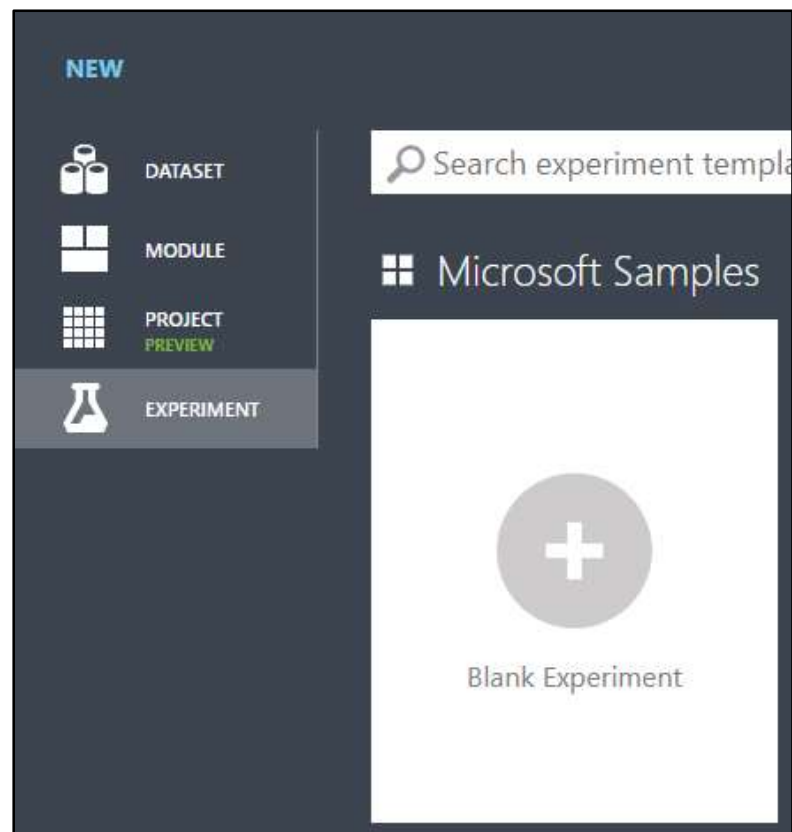
6. Select **DataSet** > **From Local File**.



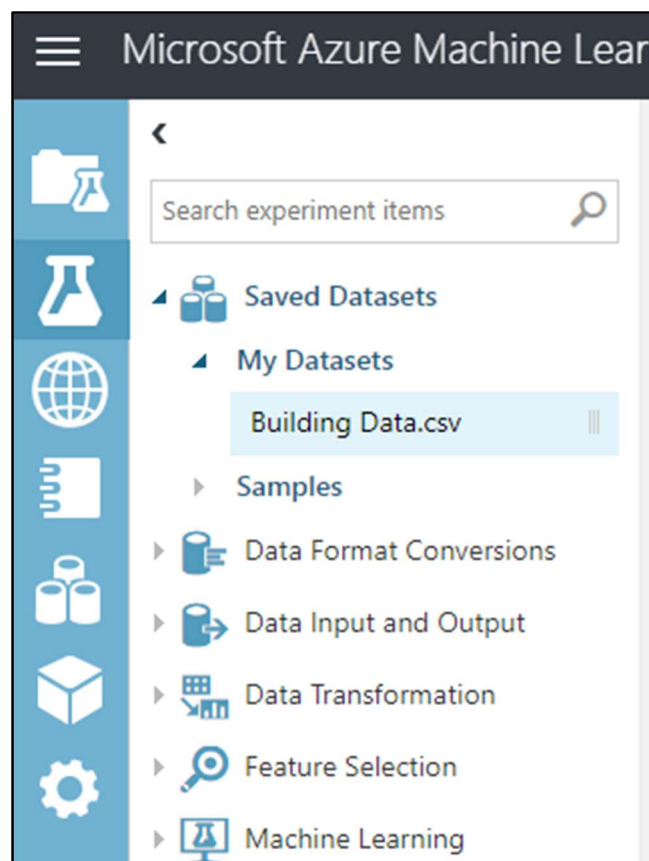
7. Select **Choose File** and then select the file you just downloaded.

A light gray form titled 'Upload a new dataset'. It contains the following fields: 'SELECT THE DATA TO UPLOAD:' with a button 'Escolher arquivo' and the text 'Building Data.csv'; a checkbox 'This is the new version of an existing dataset'; 'ENTER A NAME FOR THE NEW DATASET:' with a text box containing 'Building Data.csv'; 'SELECT A TYPE FOR THE NEW DATASET:' with a dropdown menu showing 'Generic CSV File with a header (.csv)'; and 'PROVIDE AN OPTIONAL DESCRIPTION:' with a large empty text area. A circular checkmark button is in the bottom right corner.

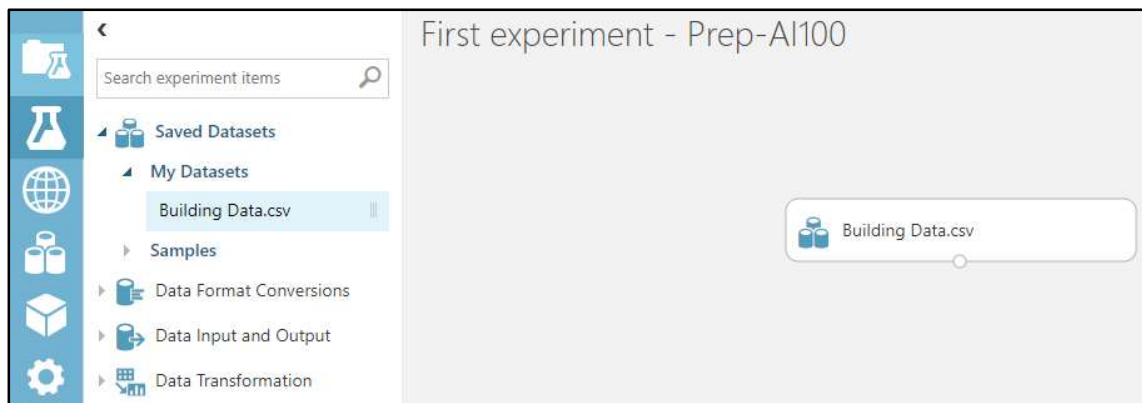
8. In the lower-right corner, select the check mark to upload the dataset.
9. Verify that **Building Data.csv** is listed as a dataset.
10. In Machine Learning Studio, select the **Experiments** icon.
11. Select **New > Blank Experiment**.



12. Select **Saved Datasets** > **My Datasets** > **Building Data.csv**.



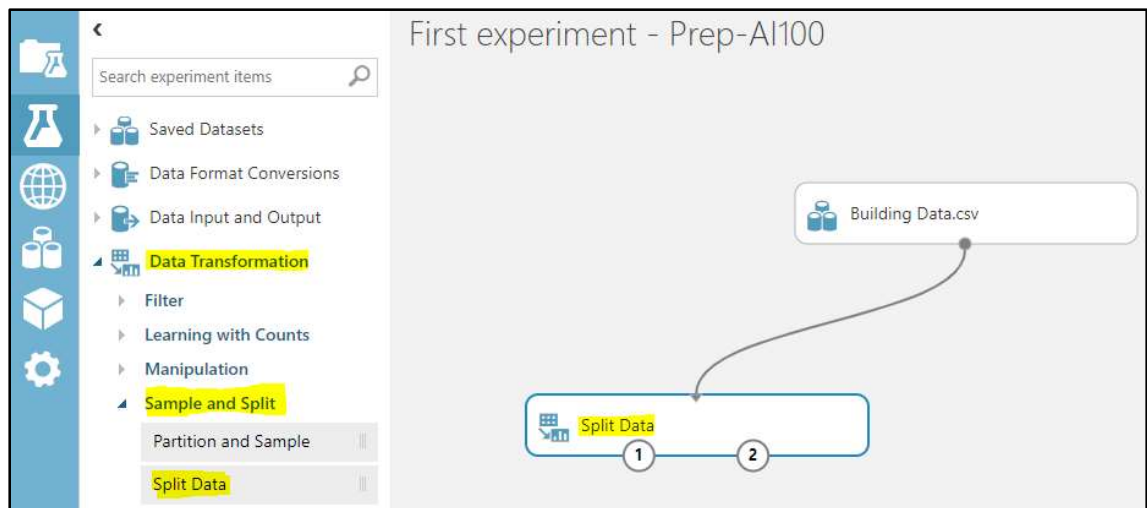
13. Drag this dataset onto the designer surface.



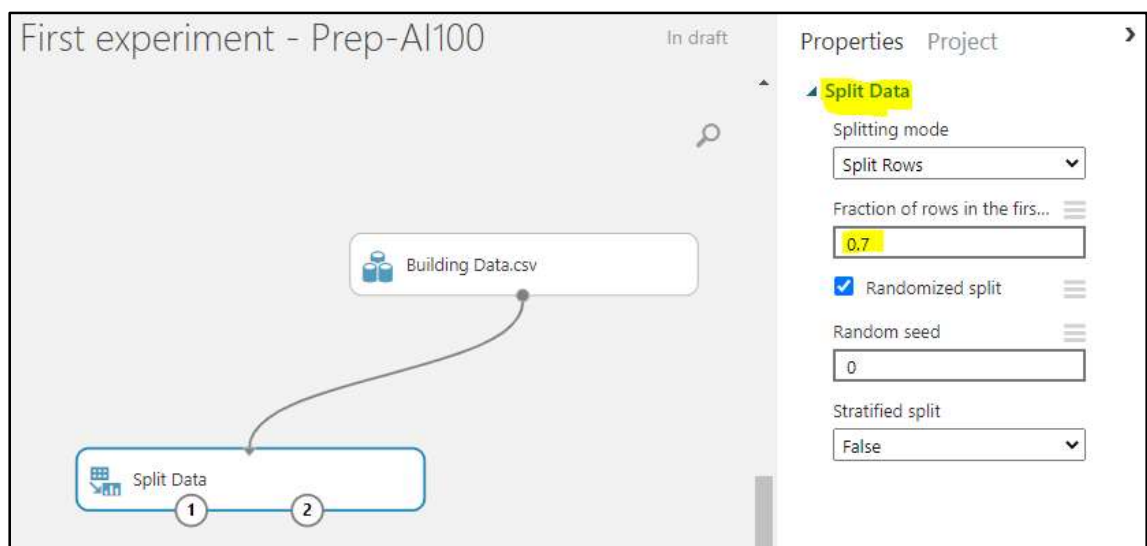
Exercise - Build the experiment

Now let's build and run the experiment.

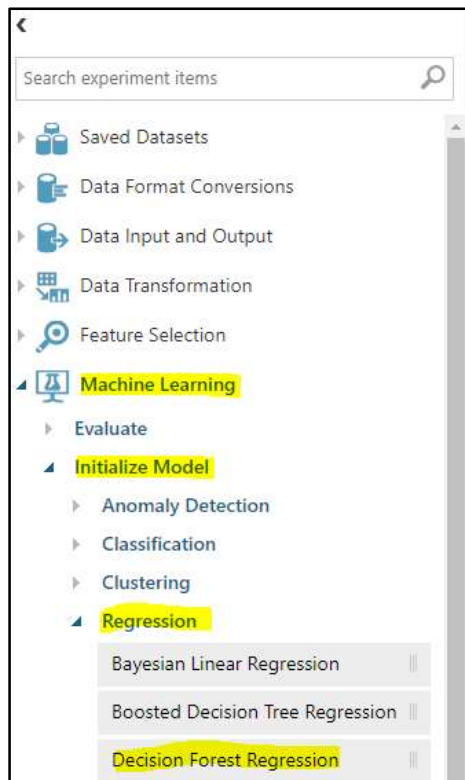
1. Expand **Data Transformation** → **Sample and Split**.
2. Drag a **Split Data** operation onto the designer surface.



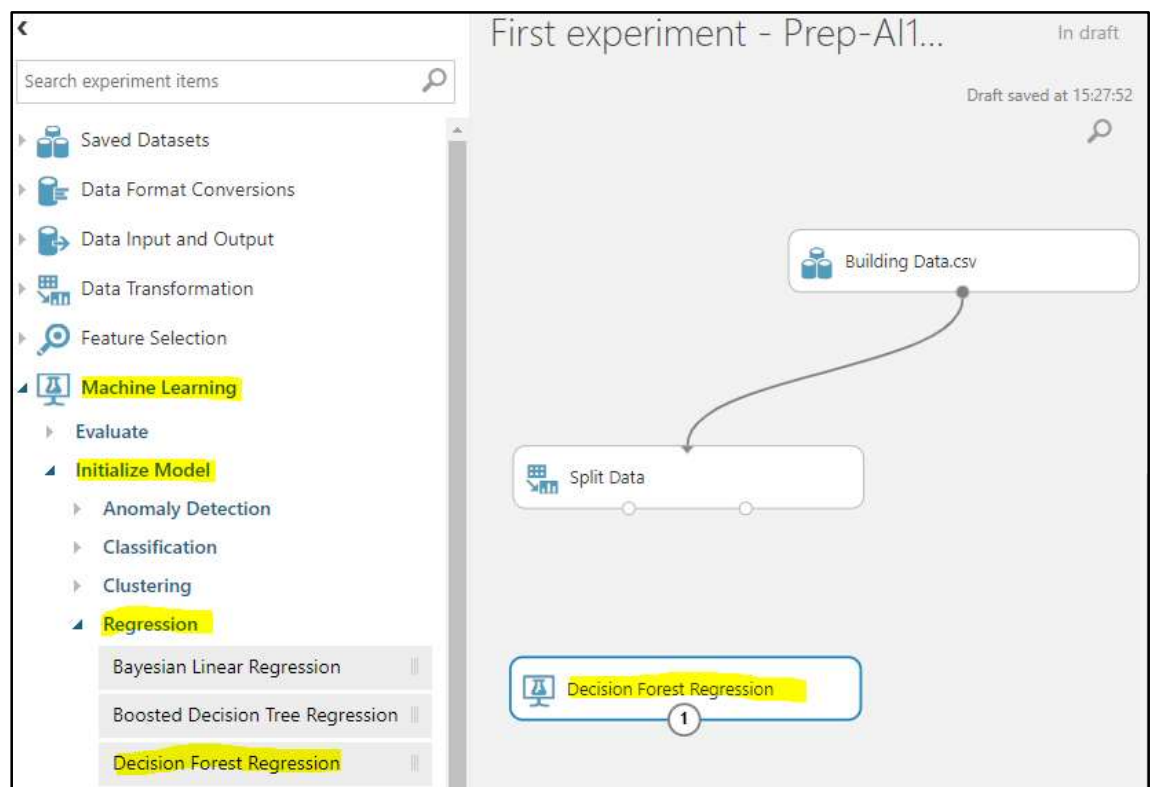
3. On the right, set the fraction of rows in the first output dataset to **0.7** to split the data at 70/30.
4. Connect the dataset to the Split Data operation by dragging the dataset handle to the Split Data operation.



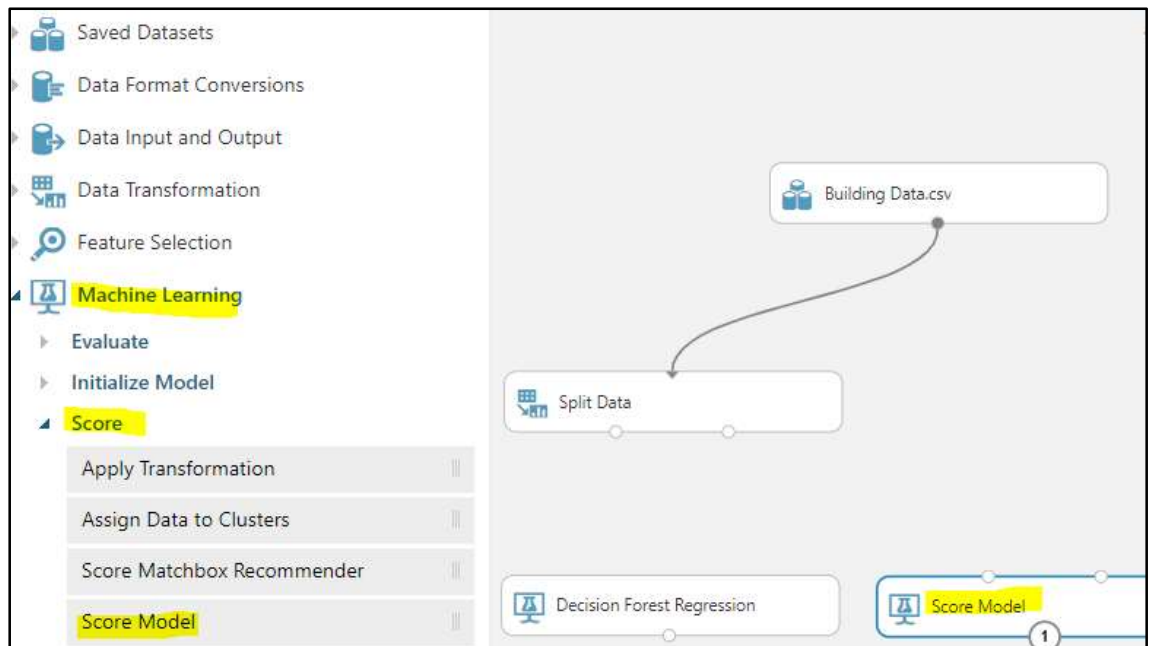
5. On the left, expand **Machine Learning** → **Initialize Model** → **Regression**.



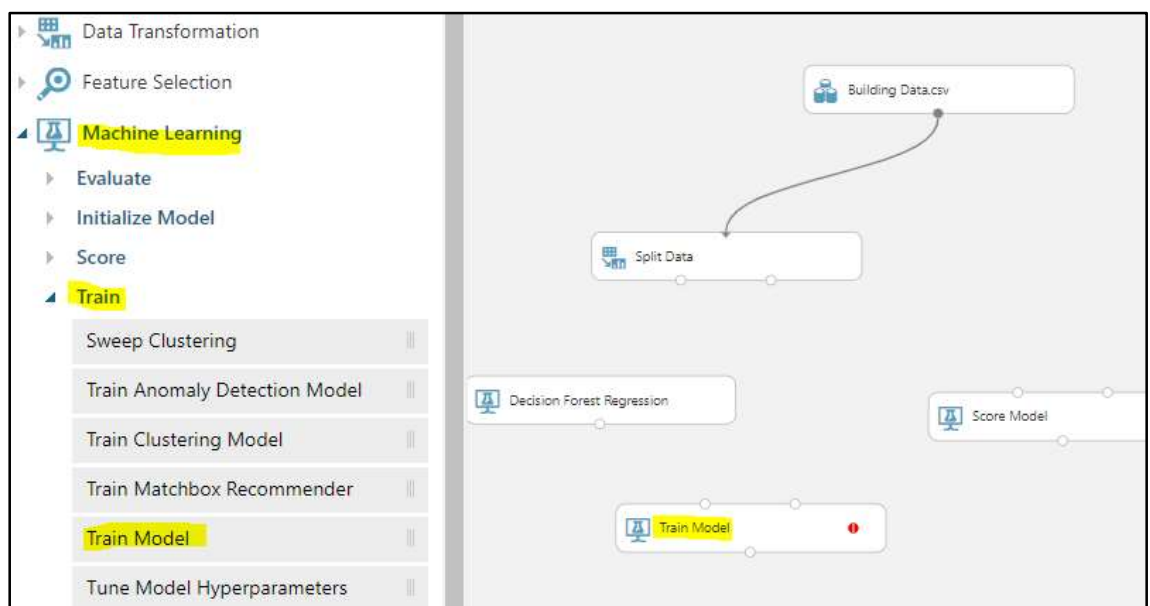
6. Drag the **Decision Forest Regression** entry onto the designer surface.



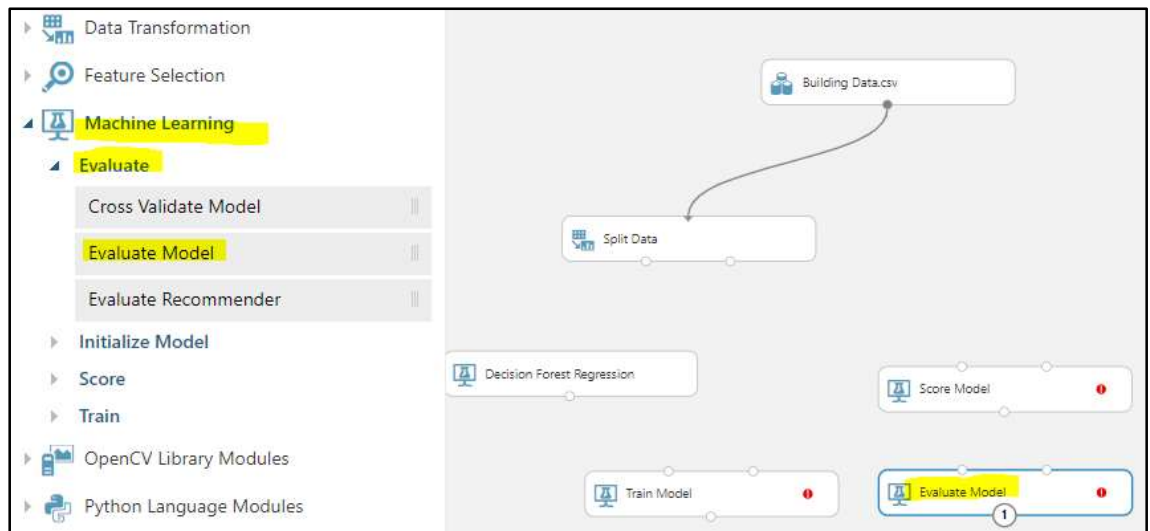
7. On the left, expand the **Score** subcategory and locate **Score Model**.
8. Drag the **Score Model** operation onto the designer.



9. On the left, expand **Train** and drag a **Train Model** operation onto the designer.



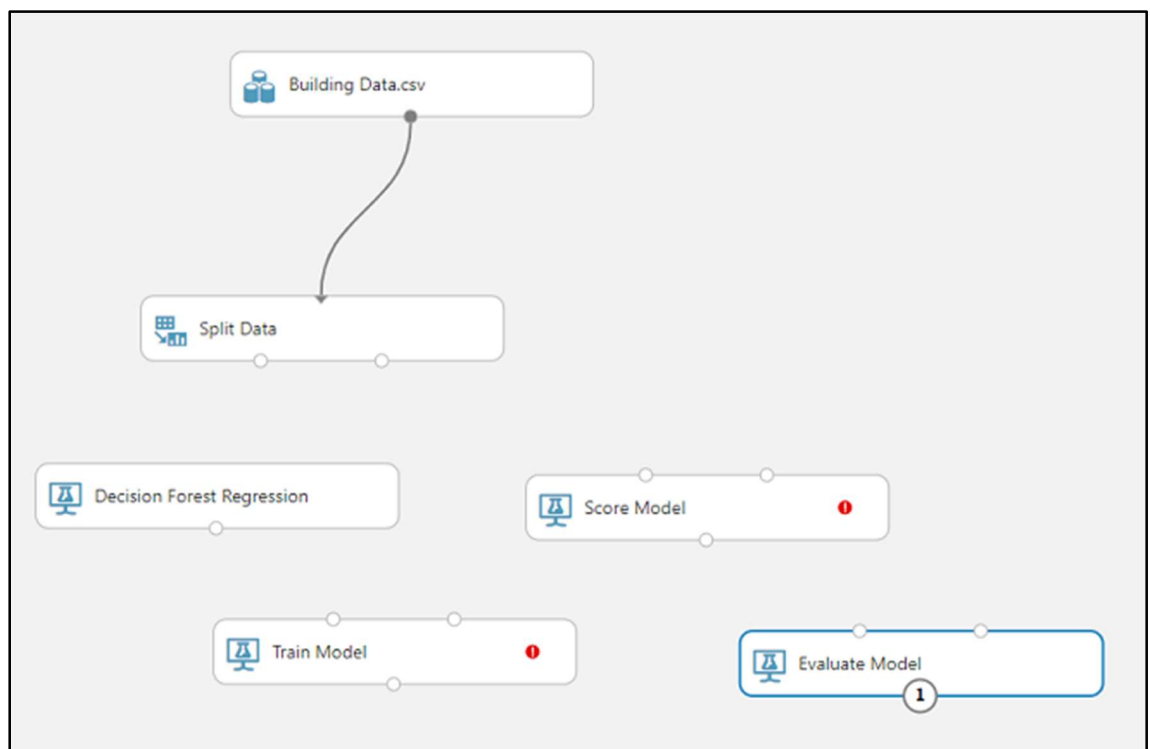
10. On the left, expand the **Evaluate** subcategory of Machine Learning and drag an **Evaluate Model** operation onto the designer.



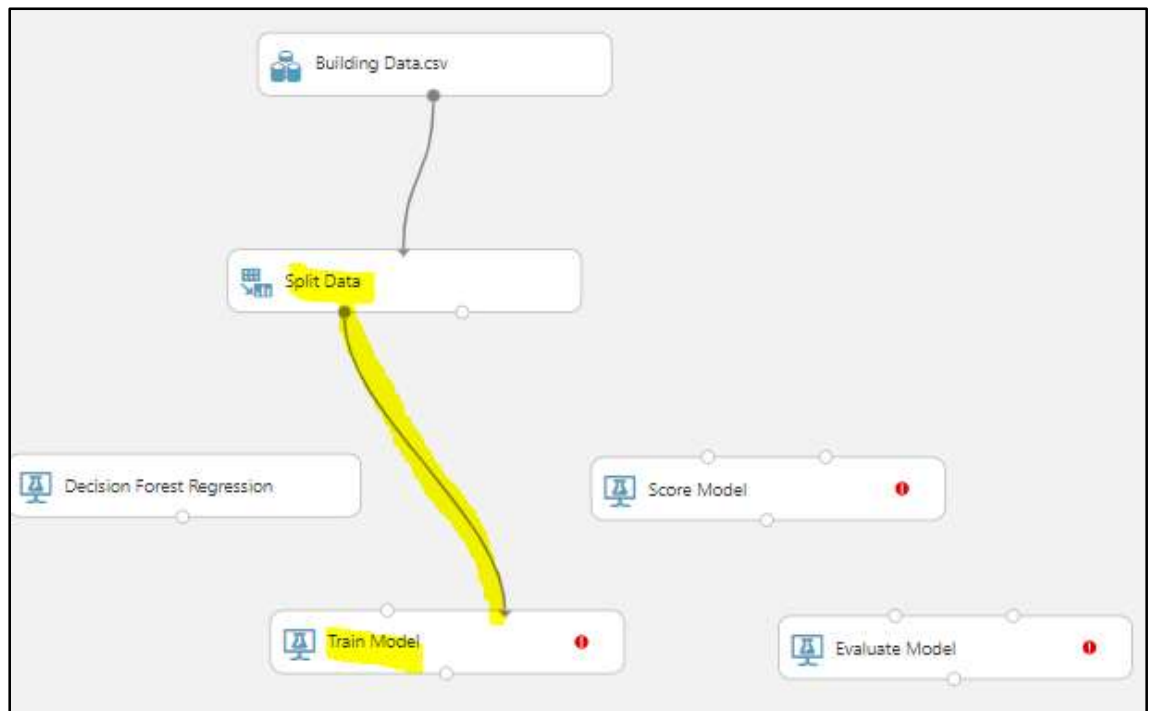
Tip

If you run out of space on the designer surface, you can scroll the designer surface area or use your mouse wheel to zoom in or out.

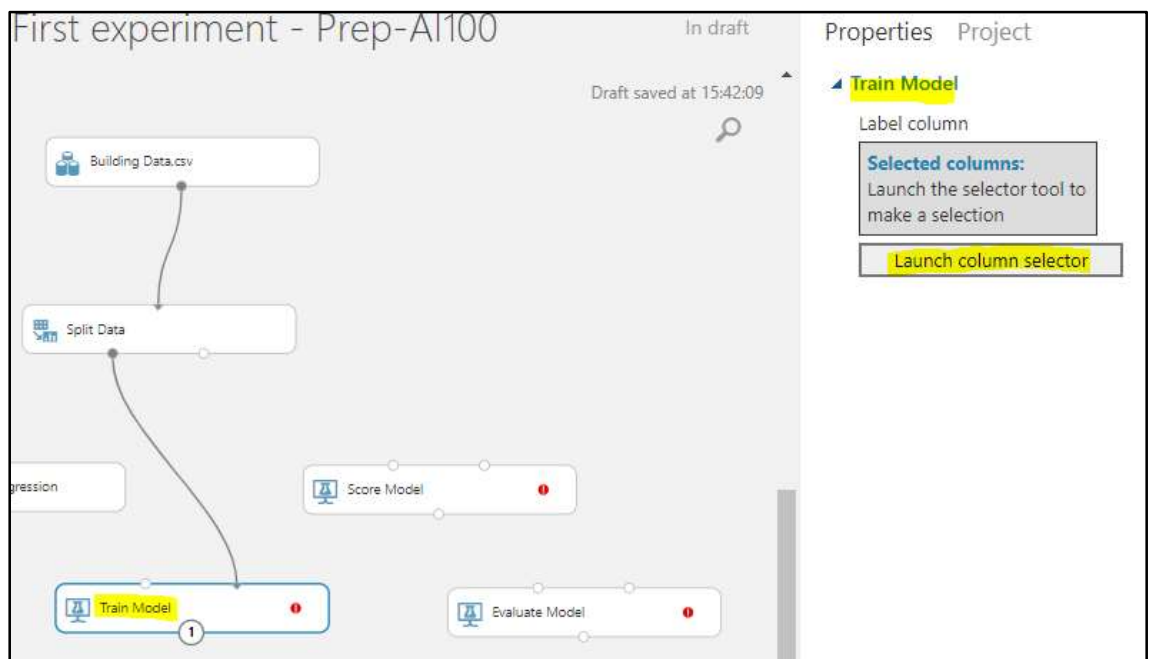
Your experiment window should now look like this:



11. Connect output 1 of the **Split Data** operation to the **Train Model** operation.



12. Select the **Train Model** operation.
13. On the right, select **Launch Column Selector**.




14. Select the **With Rules** entry. Then click inside the empty column list and select the **Wall Area** column name.
15. Select the check mark to accept the selection.

Select a single column

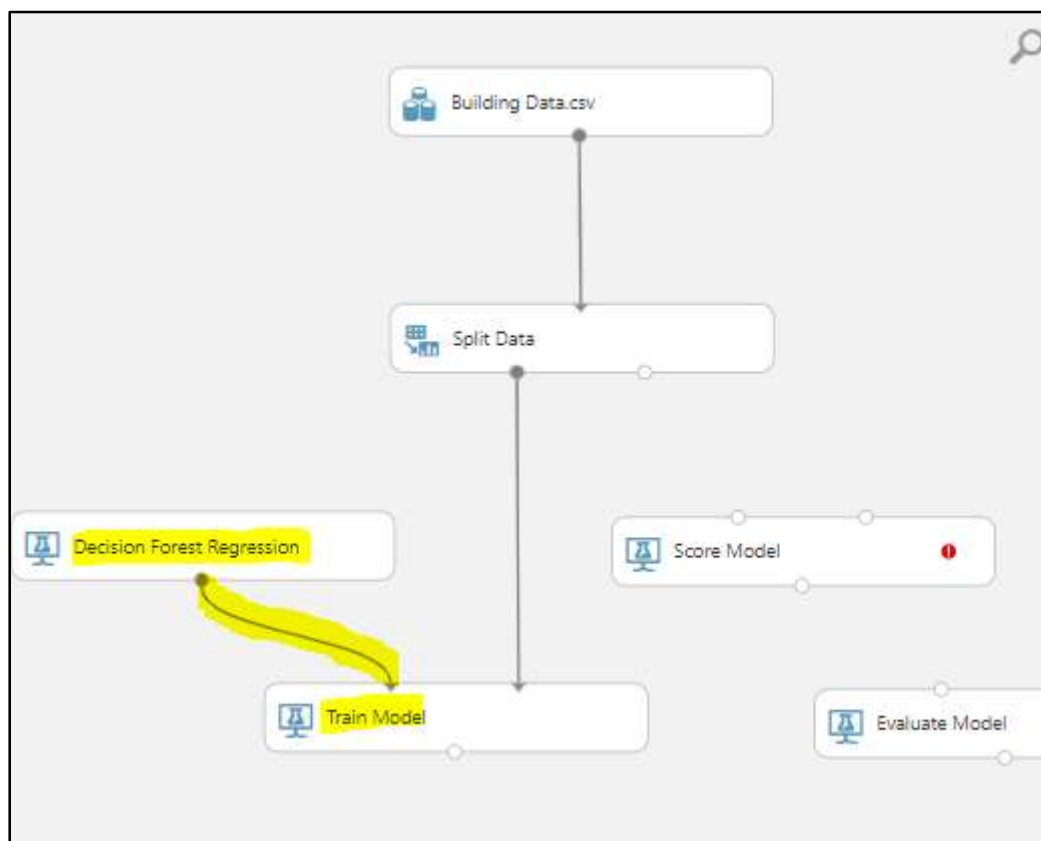
BY NAME
WITH RULES

Include column names

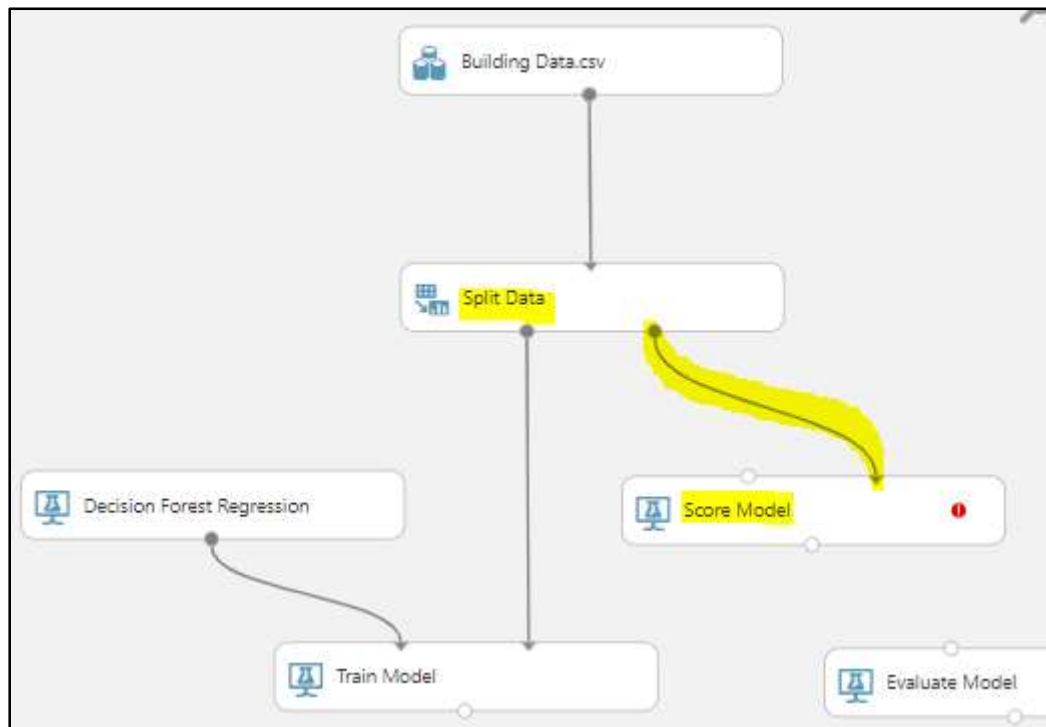
WallArea ✕



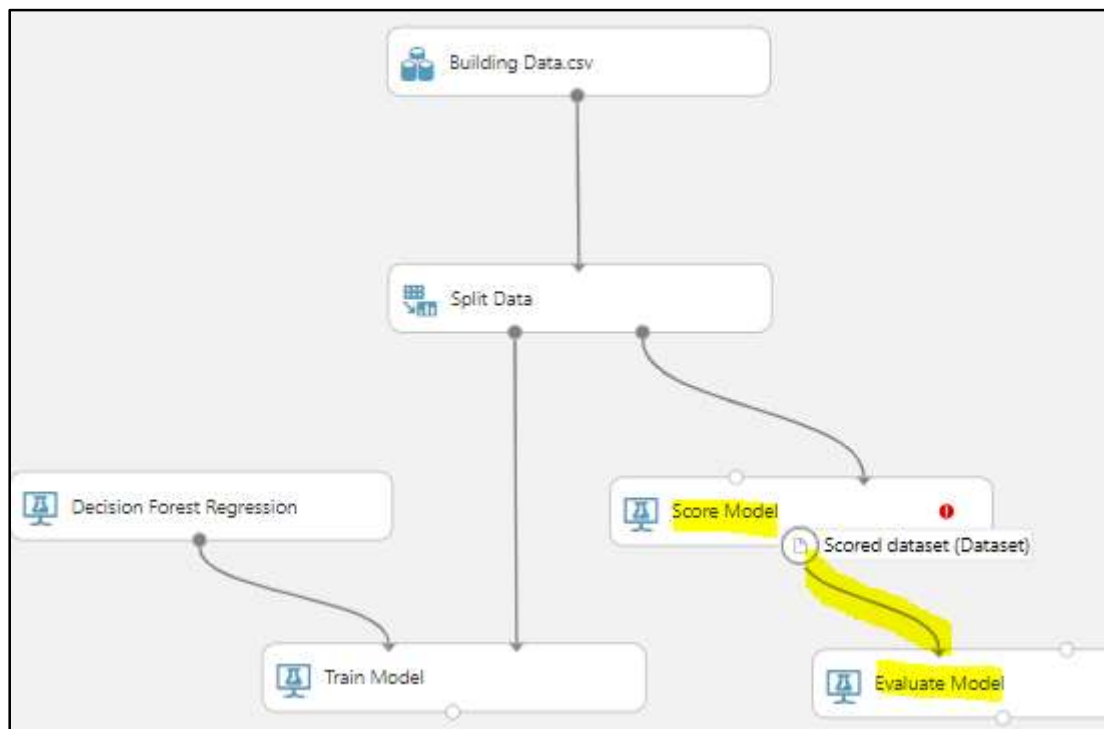
16. Connect the **Decision Forest Regression** operation to the remaining input of the **Train Model** operation.



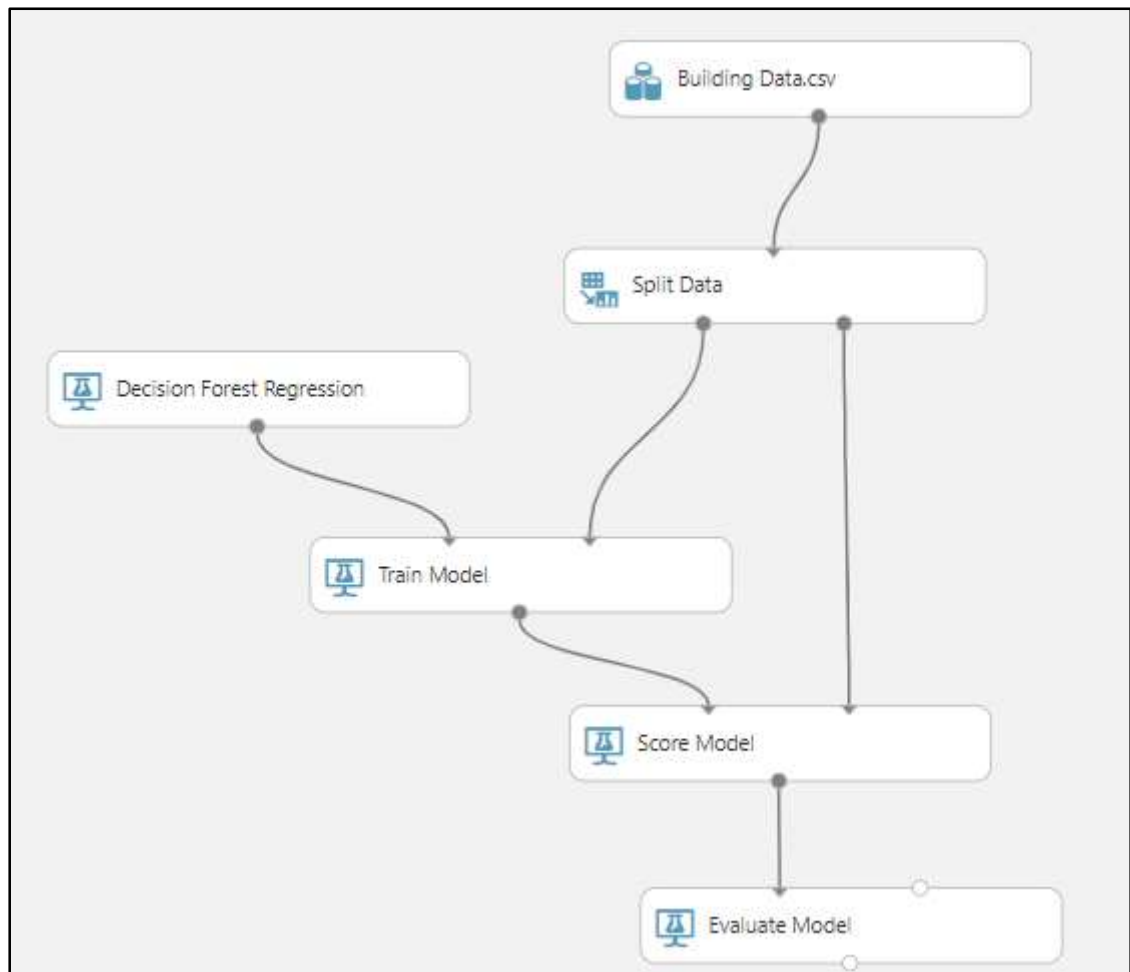
17. Connect the second output of the **Split Data** operation to the **Score Model** operation.



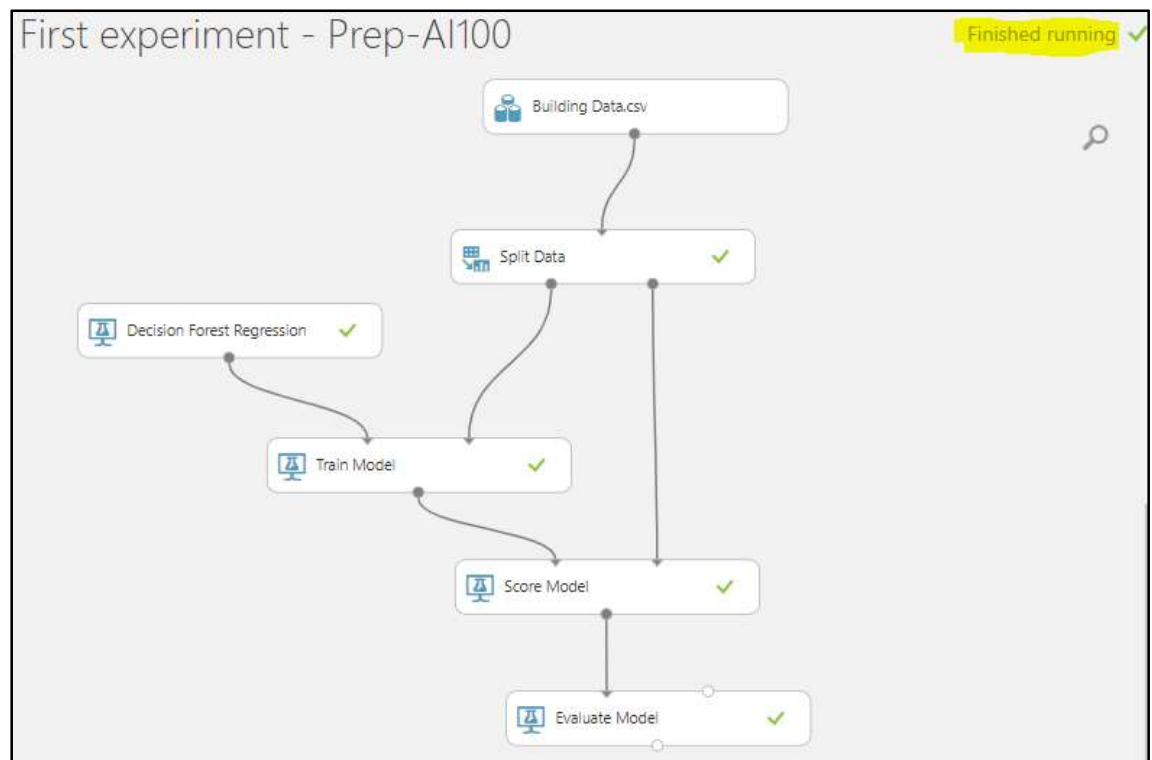
18. Connect the output of the **Score Model** operation to the left input of the **Evaluate Model** operation.



19. Connect the output of the **Train Model** operation to the remaining **Score Model** input. Your experiment should now look like this:



20. At the bottom of the designer window, run the experiment by selecting the **Run** button.
21. Make sure all of the operation boxes show green check marks and that the upper-right corner of the designer displays **Finished running** with a green check mark.



22. Save your experiment as **Energy Efficiency Regression**.

At this point, feel free to look at the data in the CSV file to figure out what your experiment is acting on.

Exercise - Prepare and deploy the experiment

You've created and trained an experiment in Azure Machine Learning Studio. Now it's time to publish it as a web service that clients can access.

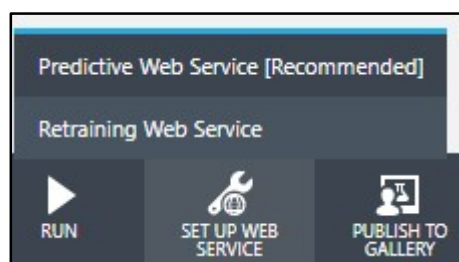
Let's prepare and publish your experiment as a web service and then evaluate the results.

You should have already created the energy-efficiency experiment that you'll publish as a web service. Here are your next steps:

1. On the left navigation bar of Machine Learning Studio, select the **Experiments** icon.
2. Select your completed **Energy Efficiency Regression** experiment to open it.
3. Depending on the state of the web service, you might not have to run your experiment again. If the upper-right corner of the designer displays **Finished running**, and your stages show green check marks, you're ready for the next step. If not, select the **Run** button.

Set up the web service

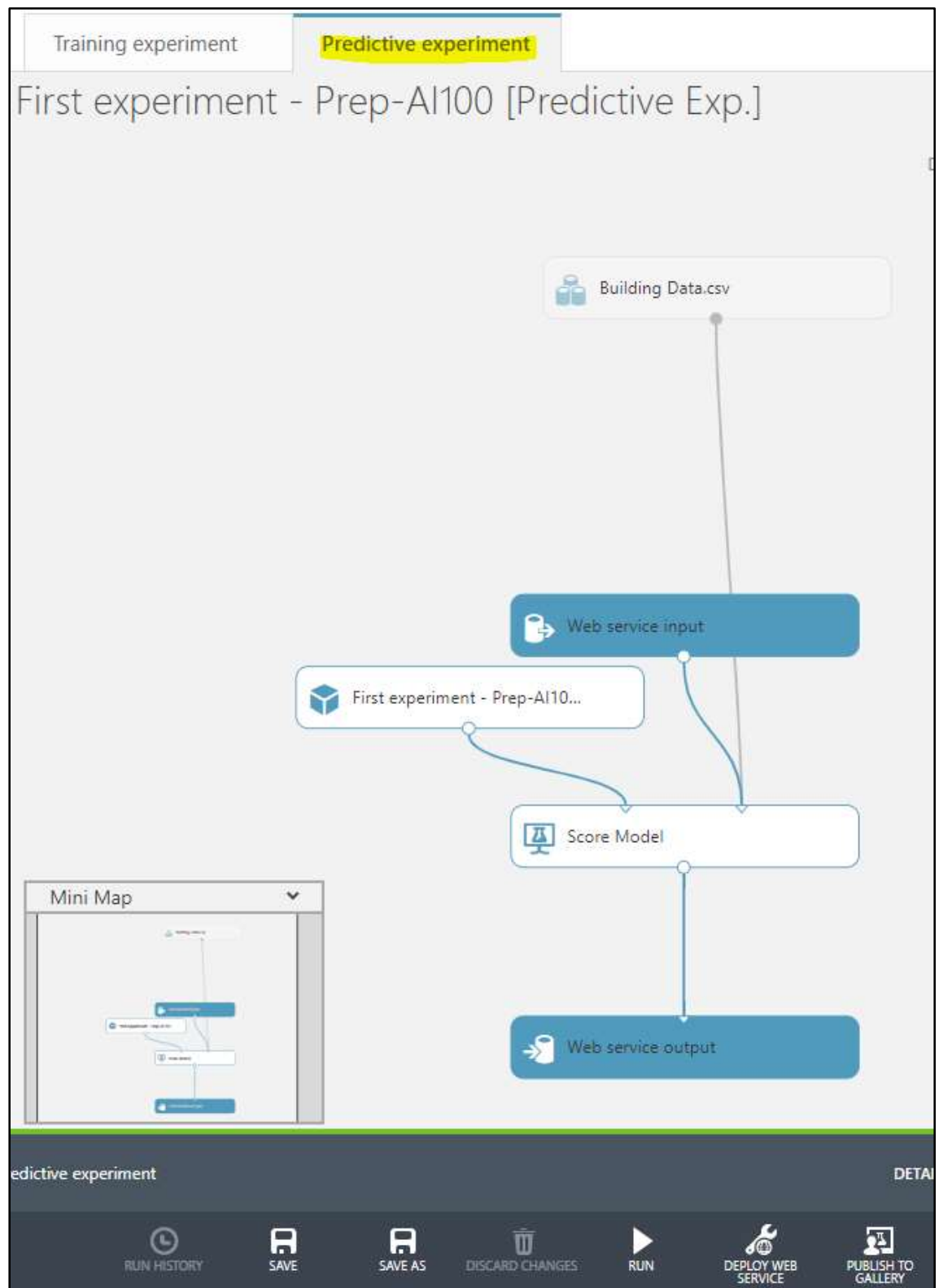
1. After the energy-efficiency experiment has run, at the bottom of the designer, hover over the **Set Up Web Service** button.
2. Select **Predictive Web Service** [Recommended].



Tip

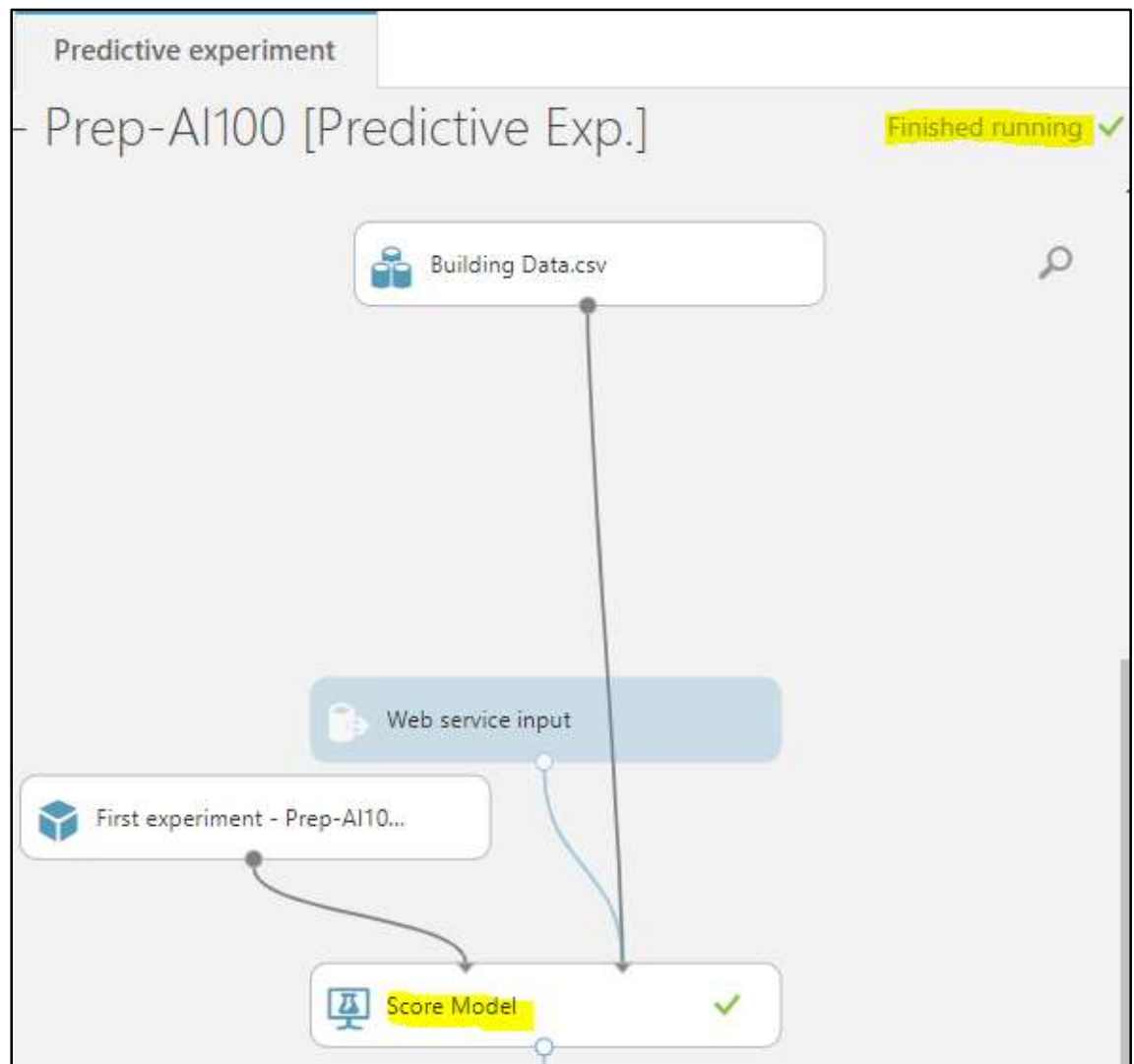
If the **Predictive Web Service** option is not available, run your experiment again. Check for the **Finished** running note.

3. Verify that the **Predictive experiment** tab opens. The designer should show a web service input and web service output operation.



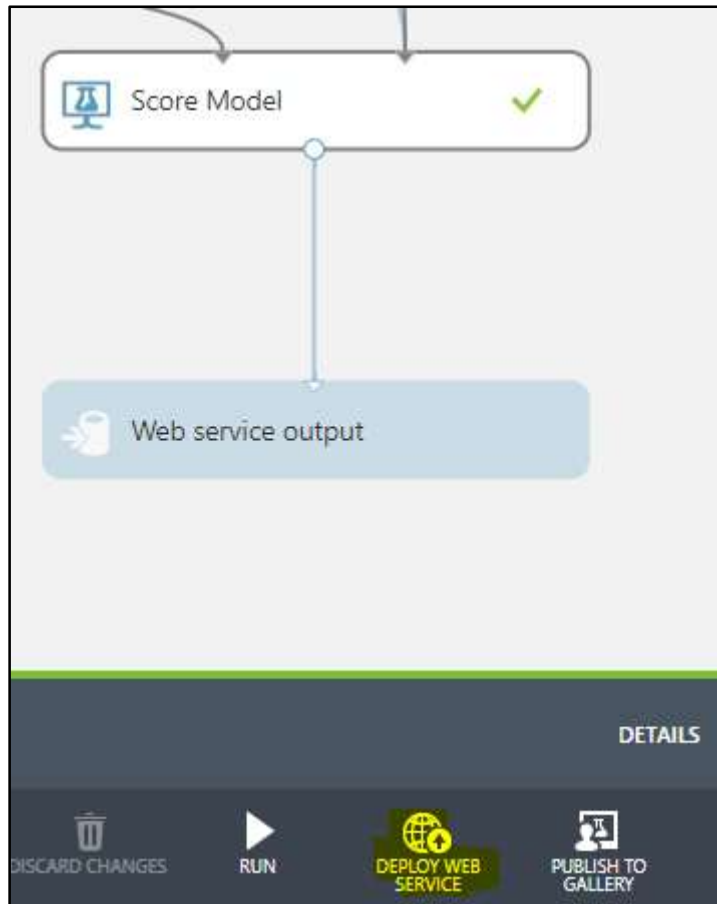
4. Select the **Run** button.

5. Verify successful completion by looking for a green check mark in the **Score Model** operation. Also look for a **Finished running** note in the upper-right corner.



Deploy the web service

Select the **Deploy Web Service** button. Machine Learning Studio creates the web service and opens a new window that indicates success. You'll notice an API key for your web service. Applications will use this key to access the web service.



Microsoft Azure Machine Learning Studio (classic) Web Services

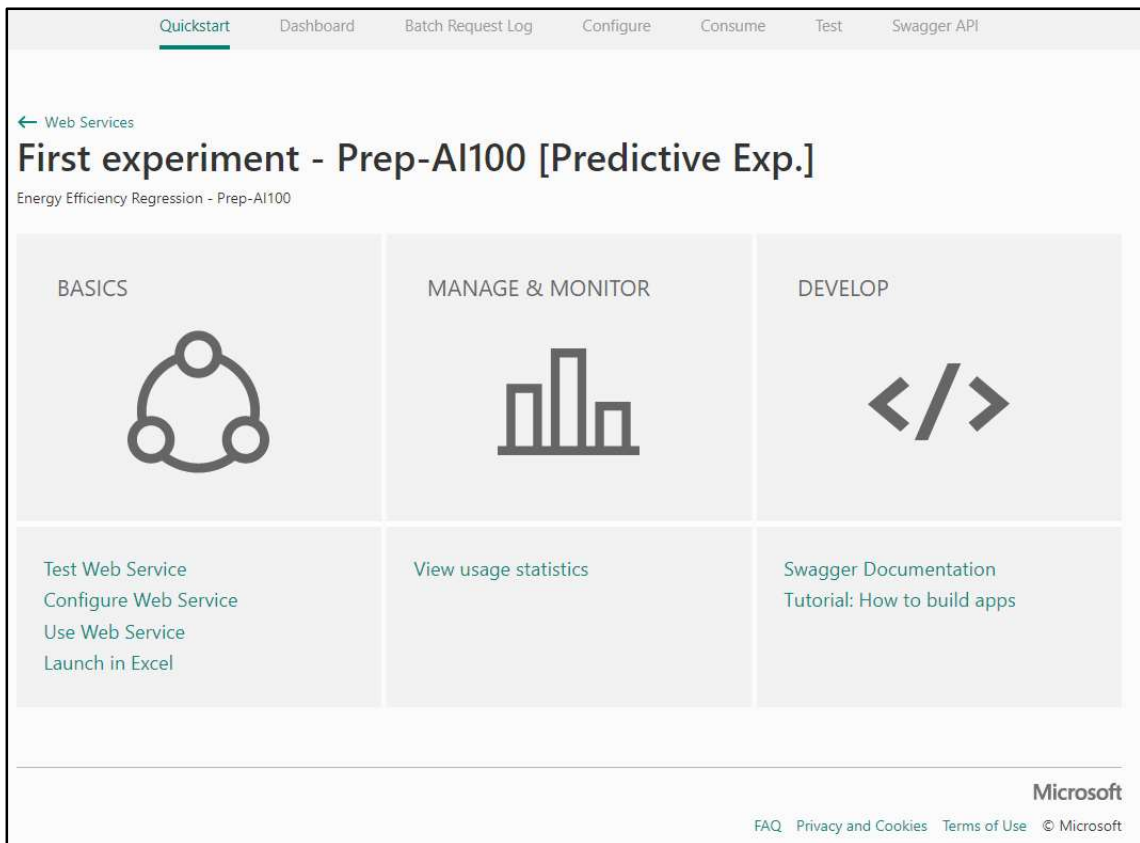
This site uses cookies for analytics, personalized content and ads. By continuing to browse this site, you agree to this use. [Learn more](#)

Deploy "First experiment - Prep-AI100 [Predictive Exp.]" experiment as a web service

Web Service Name	Energy_Efficiency_Regression_Prep-AI100
Storage Account	<div></div> <p>The storage account shown is used by the workspace. The same storage account will be used by the web service.</p>
Price Plan	<div></div>

Important: The plan tiers default to the plans in your default region and your web service will be deployed to that region.
By clicking on "Deploy", you agree to pay the plan charges in accordance with the [Pricing Page](#).

Processing...



Test the web service

Now that the web service is deployed, let's test it to see the output results.

1. Next to the **Request/Response** option, select the **Test** button to open the data entry dialog box.
2. In the data entry dialog box, enter the following values from the training model data:
 - Wall area: **296**
 - Roof area: **110.25**
 - Overall height: **7**
 - Glazing area: **0**
 - Heating load: **15.55**

DashboardBatch Request LogConfigureConsumeTest

← Web Services

First experiment - Prep-AI100
[Predictive Exp.]

Energy Efficiency Regression - Prep-AI100

Request-ResponseBatch

input1

WallArea296

RoofArea110.25

OverallHeight7

GlazingArea0

HeatingLoad15.55

Test Request-Response

output1

Your prediction results will display here.

3. In the lower-right corner, select the check mark.

You'll see the output on the status bar, just above the **New** button at the bottom of the Machine Learning Studio window.

Request-Response Batch

input1

CSV

WallArea	296
RoofArea	110.25
OverallHeight	7
GlazingArea	0
HeatingLoad	15,55

Test Request-Response

output1

WallArea	296
RoofArea	110.25
OverallHeight	7
GlazingArea	0
HeatingLoad	1555
Scored Label Mean	0.571428571428571
Scored Label	0.000249999999684405
Standard Deviation	

To see an example of the returned JSON code, select the **Details** button. You'll see the values you entered and two additional values. These are the model's evaluations of the energy efficiency for the values you entered.

Exercise - View headers and sample code

Clients consume your experiment's web service by using RESTful API calls. Azure Machine Learning Studio gives you some sample header information that shows how to call the web service.

View the request and response headers

1. Under **API Help Page for the Default Endpoint**, select the **Request/Response** link. Machine Learning Studio opens another browser tab or window to show the API documentation for this web service.
2. Scroll down to **Request Headers** and **Request Body** to get an idea of how the request will be formed. This request will be sent to the web service for evaluation. It's in JSON format.
3. Scroll down more and evaluate the **Response Headers** and **Response Body** sample. This section shows the format of the data that will be returned along with a sample JSON response. Use this sample to understand how to parse the JSON code for the data you want to use in your application.
4. Scroll down more to find information about the input and output parameters. These parameters include the expected names and data types. If you create a class to represent the returned information, you can use these parameters to create the proper data types for your member variables. The last two returned items map a **Scored Label Mean** value to a **Scored Label Standard Deviation** value.

Evaluate the sample code

You'll use the sample code as a reference or basis to create a client application that will consume the web service for the experiment. Review the code to see how to create the request and consume the response in your app.

Tip

The **Sample Code** entry contains samples in C#, Python, and R.

To view the sample code:

1. Scroll all the way to the bottom of the web service API documentation window.
2. Select the tab of the language you're most familiar with. Then review the code to see what it's doing.
3. Notice the placeholder for the API key. You'll need to replace the value **abc123** with a valid API key.

Sample Code Request-Response with Python

```
23 url = 'https://ussouthcentral.services.azureml.net/subscriptions/d4ce52bbe3d44170ad4cf143f1e8494e/services/1866c3d316eb40e8a82e7ec550e8deb5/execute?api-version=2.0&format=swagger'
24 api_key = ' ' # Replace this with the API key for the web service
25 headers = {'Content-Type': 'application/json', 'Authorization': ('Bearer '+ api_key)}
26
27 req = urllib.request.Request(url, body, headers) # Python 3
28
29 try:
30     response = urllib.request.urlopen(req) # Python 3
31     result = response.read()
32     print(result)
33 except urllib.error.HTTPError as error: # Python 3
34     print("The request failed with status code: " + str(error.code))
35
36     # Print the headers - they include the request ID and the timestamp, which are useful for debugging the failure
37     print(error.info())
38     print(json.loads(error.read()))
39
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE 1: powershell

```
PS D:\Documents\AI-100\Azure Machine Learning> py .\Sample Code Consume API Web Service.py
b'{"Results":{"output1":[{"WallArea": "329", "RoofArea": "120.57", "OverallHeight": "8", "GlazingArea": "1", "HeatingLoad": "17.26", "Scored Label Mean": "0.5", "Scored Label Standard Deviation": "0.0714289214277134"}]}}'
```

```
PS D:\Documents\AI-100\Azure Machine Learning> py .\Sample Code Consume API Web Service.py
b'{"Results":{"output1":[{"WallArea": "329", "RoofArea": "120.57", "OverallHeight": "8", "GlazingArea": "1", "HeatingLoad": "17.26", "Scored Label Mean": "0.5", "Scored Label Standard Deviation": "0.0714289214277134"}]}}'
```

Knowledge check

Check your knowledge - Create an experiment

1. You're trying to set up a web service for your experiment, but the button isn't available. Why?

- ☐ You can't publish this type of experiment as a web service. Instead, publish the experiment as an Azure function.
- ☐ You need to run the experiment.
- ☐ You didn't create a virtual machine to host the web service.

2. To test the web service with a client application, what do you need to add to the sample code?

- ☐ The API key.
- ☐ The experiment name. Add this name in the JSON code that's generated in the sample app.
- ☐ Your developer account credentials.

Knowledge check

3 minutes

✓ 200 XP

Check your knowledge - Create an experiment

1. You're trying to set up a web service for your experiment, but the button isn't available. Why?

- ☐ You can't publish this type of experiment as a web service. Instead, publish the experiment as an Azure function.

☒ You need to run the experiment. ✓

If you haven't run the experiment or if you come back to the experiment after you've signed out, you must run the experiment and get the green check marks that indicate satisfactory completion.

- ☐ You didn't create a virtual machine to host the web service.

2. To test the web service with a client application, what do you need to add to the sample code?

☒ The API key. ✓

Correct. The sample code uses **abc123** as a placeholder for the API key that you need to add.

- ☐ The experiment name. Add this name in the JSON code that's generated in the sample app.
- ☐ Your developer account credentials.

Summary

Congratulations! You've built a simple predictive analysis model using Azure Machine Learning Studio. Let's recap what you accomplished. You used an imported dataset ("ingest data") to create an experiment. Then you defined the logic in the visual designer and trained the model. Finally, you tested and published the experiment as a web service. This was a simple example, but the steps you went through are common to any AI solution you build.

Developing a model like this is an iterative process. As you modify the various functions and their parameters, your results converge until you are satisfied that you have a trained, effective model. You can then link this model to other advanced AI services in Azure, or use this solution as part of a bigger application structure to meet the business needs.

Learn more

- [Azure Machine Learning Studio documentation](#)
- [Azure Machine Learning Studio Capabilities Diagram \(PDF\)](#)
- [Sample datasets for Azure Machine Learning Studio](#)

Learn more with a Channel 9 video series

- [ML.NET](#)