Microsoft

Cortana Intelligence Suite

Lab CIS008

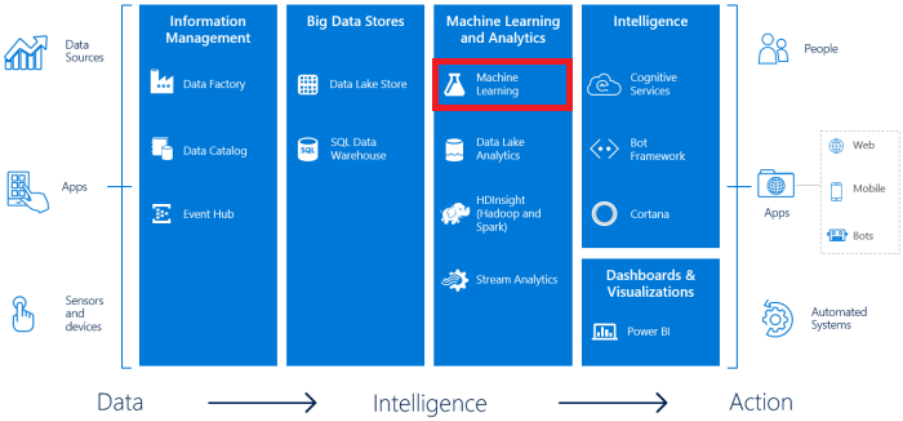
Deploying a Predictive Model with Azure Machine Learning

Contents

Overview

Summary

This lab explores unsupervised learning in Azure Machine Learning and how to deploy a predictive model as a web service.



The lab will walk through copying an experiment from the Cortana Intelligence Gallery into the ML Studio, creating a predictive experiment, deploying a model as a web service, and interacting with the API using the included web interface.

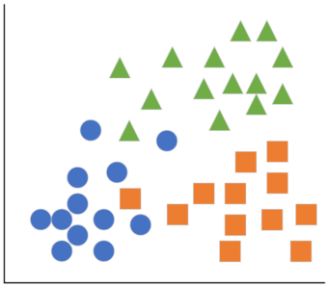
Business Case

"Where should I open my next restaurant location?"

This question is often very difficult to answer. The right choice could lead to increased revenue and profit, but the wrong choice could lead to losing a major investment. Trying to make this decision by manually sifting through hundreds or even thousands of possible cities or neighborhoods can be almost impossible. Machine learning can help with this task by analyzing large volumes of data about different locations, finding common characteristics among locations, and grouping those like-attributed locations together. These groups can then be compared to previously successful restaurant locations to help narrow the choices for where to open next.

In this lab, you will work with a dataset that includes geographic, economic, and demographic data about different US cities.

The model you will explore uses a K-Means algorithm to cluster cities into distinctive buckets. K-Means Clustering is an unsupervised learning approach that uses iterative techniques to analyze and then group together the cases with similar characteristics. These groups can then be used for labeling, exploration, identifying anomalies, or even further prediction.



Learning Objectives

Upon completing this lab, you will have hands-on experience with the following functions and concepts related to Azure Machine Learning:

- Copying an experiment from the Cortana Intelligence Gallery to the ML Studio
- Normalizing data
- Creating a Predictive Experiment
- Deploying a model as a web service
- Modifying Web service inputs and outputs
- Testing the web service via the web UI

Lab Requirements/Prerequisites

- A Microsoft account is required to access an Azure Machine Learning workspace. If you don't already have a Microsoft account, you can obtain one for free by following the link below:

  https://www.microsoft.com/en-us/account/default.aspx

Copy an Experiment from the Gallery

Access an Azure Machine Learning Workspace

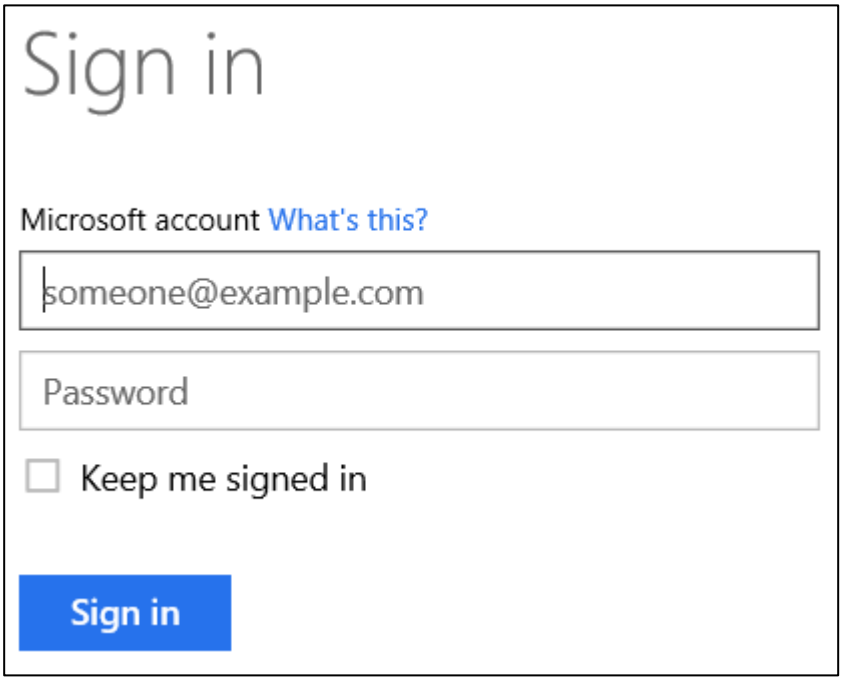
To get started, you will need to log in to a free Azure Machine Learning workspace. A workspace is like an all-inclusive development environment with the tools to create, manage, and publish machine learning models.

1. Open **Internet Explorer** or another preferred browser using the shortcut on the desktop taskbar.

2. Go to the **ML Studio** website by typing http://studio.azureml.net in the address bar.

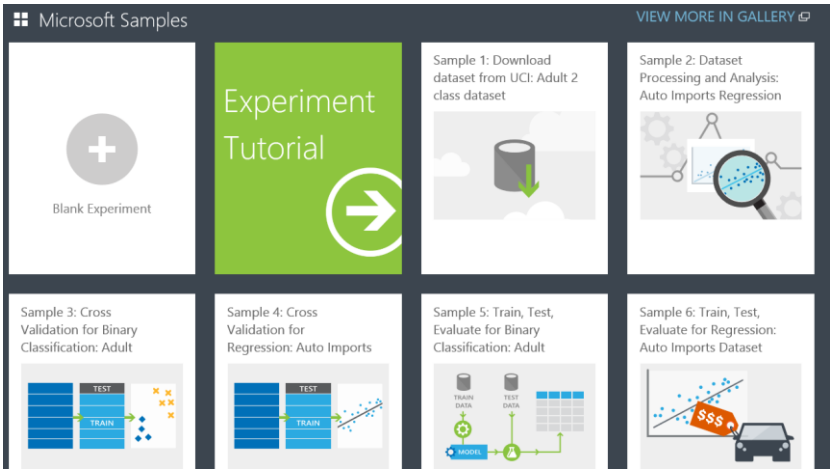
3. Click **Sign In** on the top right corner of the web page.

4. Enter the **email address** and **password** associated with your Microsoft ID, and click the **Sign In** button.

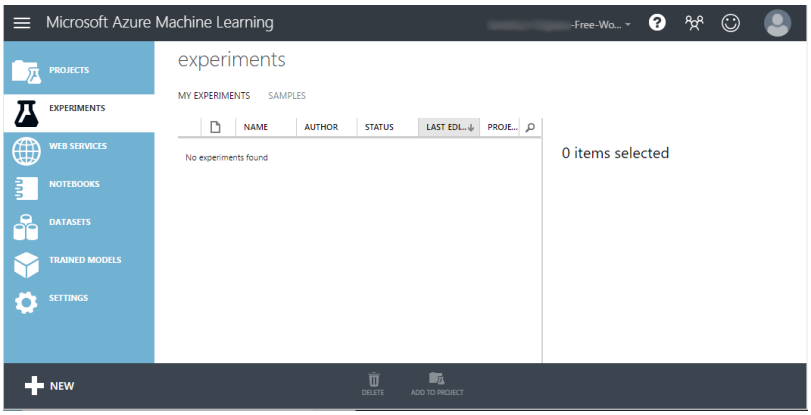


5. If upon logging in, a **Welcome** video is displayed (usually displays on the first login), click the **X** at the top right of the video to close it.

6. If the **Microsoft Samples** dialogue box is displayed (usually displays on the first login), go ahead and close it by choosing the **X** in the top right corner of the pane.



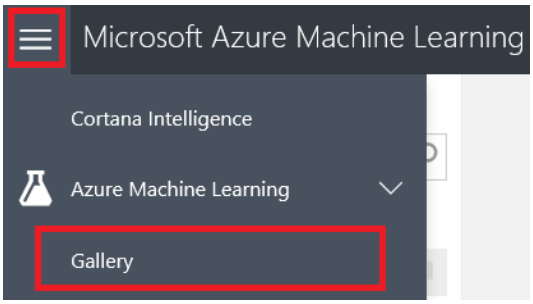
You are now logged into the free workspace associated with your Microsoft ID.



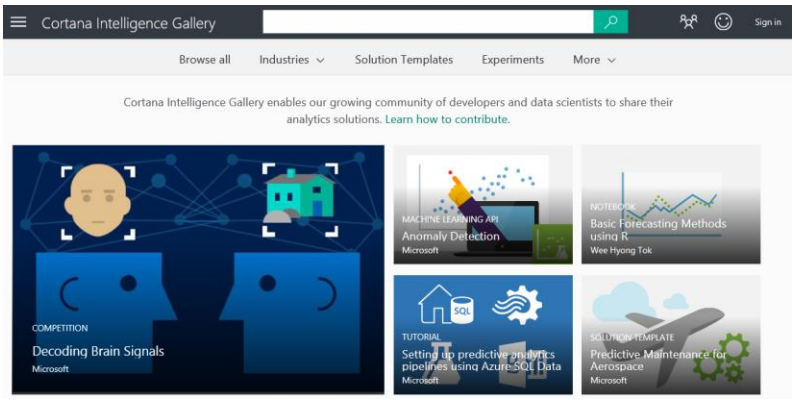
Copy Experiment from Cortana Intelligence Gallery

Next, you will go to the Cortana Intelligence Gallery to find an experiment that has already been created for you, and copy it into your workspace.

1. Click the  button on the top left corner of the workspace, and select **Gallery** from the dropdown menu.

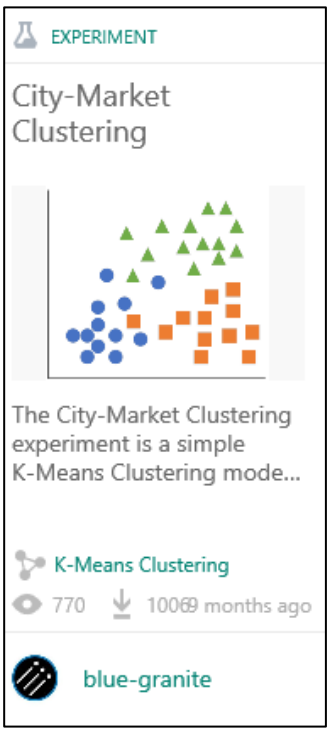


The *Cortana Intelligence Gallery* website will open. The gallery is Microsoft's community portal for the Cortana Intelligence Suite. Here, users can upload experiments and share tutorials and comments with each other.



2. In the search box near the top of the website, type **mslabs** and hit **Enter**.

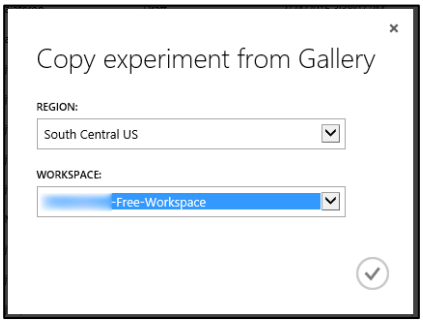
3. Find the experiment titled **City-Market Clustering** and click either the **title or picture** on the experiment.



This will open the Experiment page in the gallery. Notice the user who contributed the experiment, a *Summary* and *Description* for the experiment, and some metadata about the experiment in the panel on the right.

4. Click the **Open in Studio** button found in the panel on the right side of the screen.

5. In the **Copy experiment from Gallery** dialogue box, choose the **workspace** from the dropdown box you want to copy the experiment into (this should be the same workspace you logged into at the beginning of the lab). Use the default **Region** for this experiment.



6. Click the  button on the **Copy experiment from Gallery** dialogue box.

After a few moments, the experiment you selected in the Gallery will open in ML Studio under the workspace you selected. This is a copy of the experiment from the gallery, so you are free to run/modify it as you would like. It will also be saved as a training experiment in the workspace.

7. If prompted with a notification regarding updated modules/datasets, click the 

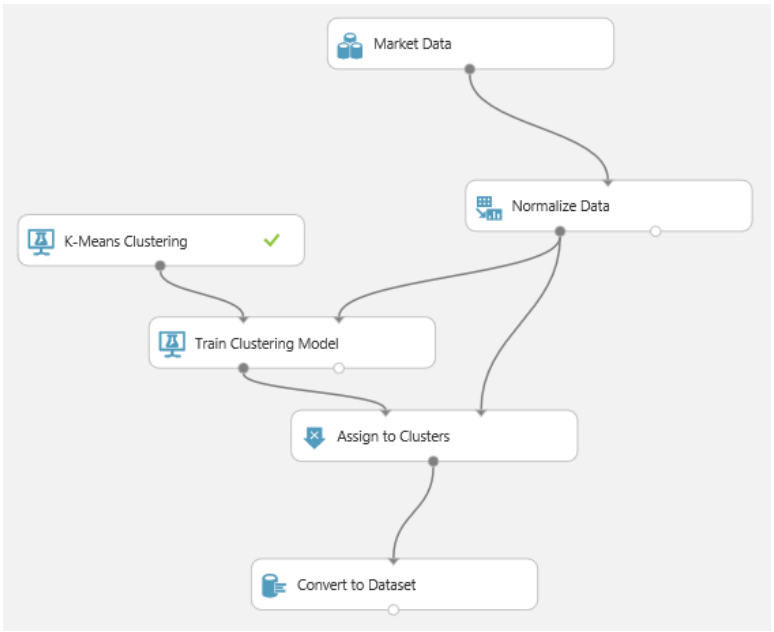
button to proceed.

8. Click **RUN** at the bottom of the page to execute the training experiment.

9. After the experiment has finished running, click the **SAVE AS** button at the bottom of the page.

10. In the **Save As** dialogue box, change the **Experiment name** to **Lab – Deploying a Predictive Model**, and click the  button.

    The ML Studio canvas should now look similar to the below image.



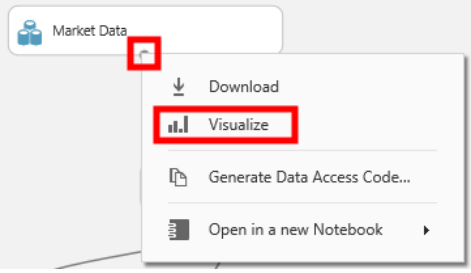
Explore the Training Experiment

Visualize the Input Dataset

Next, you will take a quick look at the dataset used for the experiment.

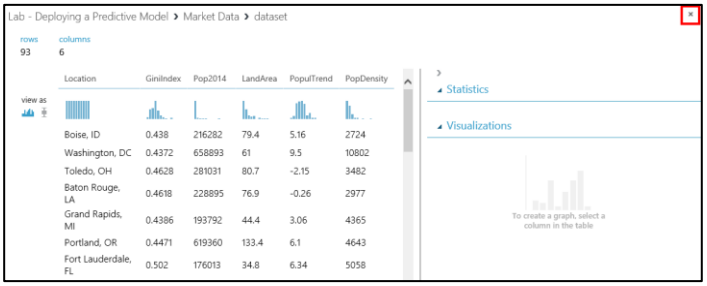
1. Click **RUN** at the bottom of the page to execute the training experiment.

2. Once the experiment finishes running, click the **output** port on the **Market Data** dataset and select **Visualize** from the displayed menu

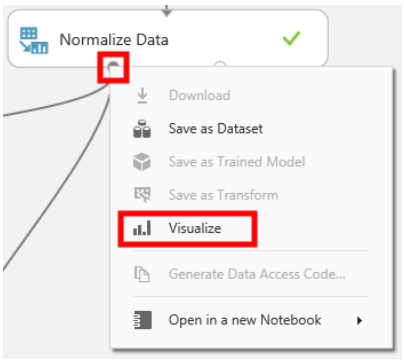


Notice each row in the data set represents a location and there are a series of geographic, economic, and demographic features describing each location.

3. Click the **X** in the top right corner of the **Visualize** dialogue box to close it.



4. Click the **left output** port on the **Normalize Data** module and select **Visualize** from the displayed menu.



Notice each of the features in the dataset has been scaled to a value between -1 and 1.

Normalizing features is the process of setting all of the numeric values in a dataset to a similar scale (usually somewhere between -1 and 1). Normalizing the data helps to level the playing field so that no single variable will overly dominate the model and typically makes features easier to visualize. Having normalized data is necessary for K-Means Clustering.

5. Click the **X** at the top right of the **Visualize** dialogue box to close it.

Explore the Clustering Model

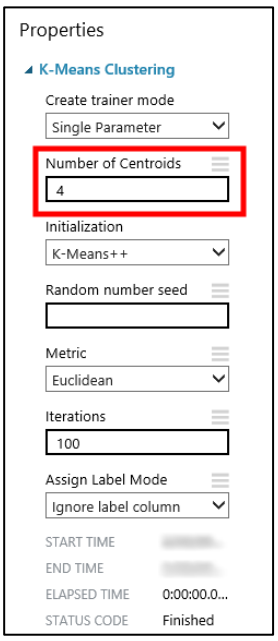
The next series of modules trains the K-Means Clustering model. Clustering is a form of *unsupervised learning*, which does not account for any prior labelling or categorization of the data. An alternative would be *supervised learning*, which could be done if the data had labels used in the experiment (such as an existing cluster assignment).

Once your K-Means Clustering module is in place, new input data is then assigned to the clusters using the trained model.

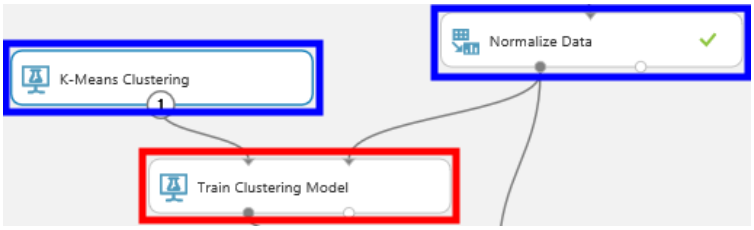
1. Click the **K-Means Clustering** module on the canvas.

Notice the different parameters that can be configured in the Properties pane, including the number of clusters (centroids) to be created.

2. In the **Properties** pane, change the value located in the **Number of Centroids** textbox to **4**.



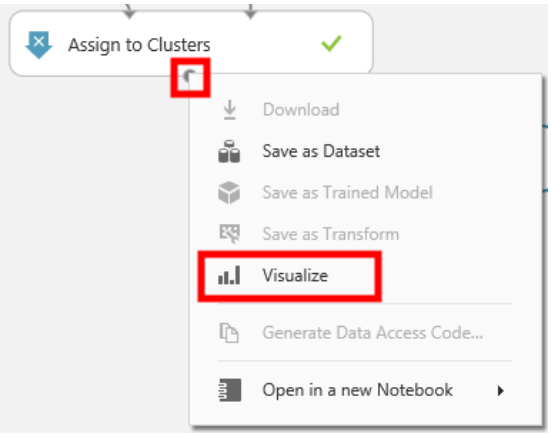
The next module in the data flow, *Train Clustering Model*, will train the clustering model using the K-Means algorithm and the input dataset.



Next, records are assigned to clusters using the *Assign to Clusters* module.

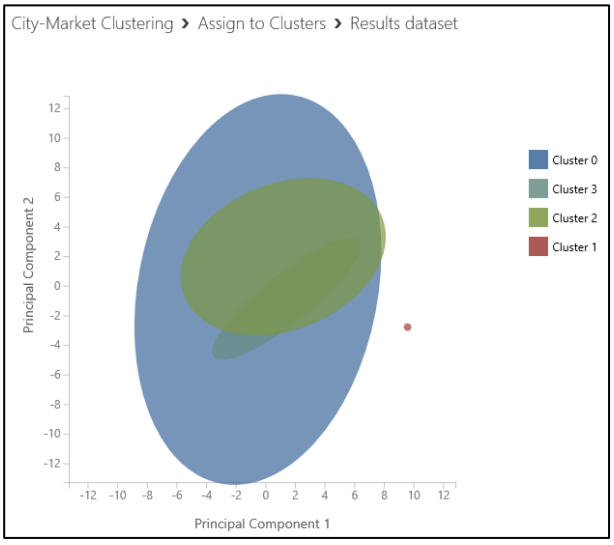
3. Click **RUN** at the bottom of the page to execute the training experiment.

4. Click the **output** port on the **Assign to Clusters** module and select **Visualize** on the displayed menu.



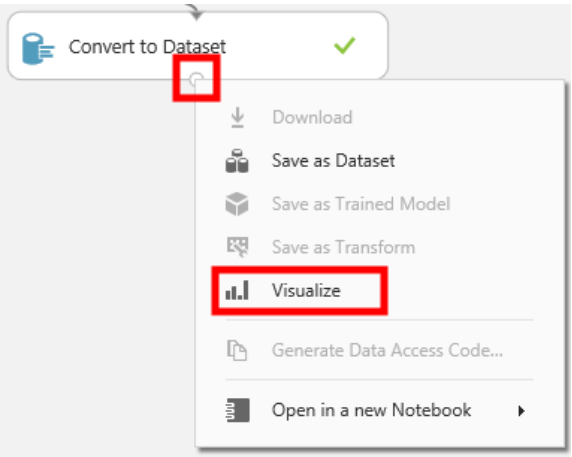
The visual that is displayed shows the size and relative position among features of each of the clusters. Your visual may differ from the screenshot displayed below.

Note that the X and Y axes reference *Principal Component 1* and *Principal Component 2*. Because the data contains multiple dimensions, it is difficult to graph in 2D. Principal component analysis offers a way to transform and then visualize the data in a way that most emphasizes variation.



5. Click the **X** in the top right corner of the **Visualize** dialogue box to close it.

6. Click the **output** port on the **Convert to Dataset** module and select **Visualize** from the displayed menu to view how each individual city-market was clustered.



The *Assignments* column has the numeric value for the assigned cluster for each row in the data set.

PopulTrend	PopDensity	Assignments
0.324862	-0.480789	0
1.563838	1.630756	0
-1.761985	-0.282652	0

This numeric value does not provide any context into what type of cities/locations are in each cluster.

At this point, you would typically download the assigned dataset to a tool like Excel to analyze and try to label each cluster. Upon doing this, you would likely find clusters for average/established cities, quickly growing cities, extremely large and dense cities, and cities that are very geographically spread out.

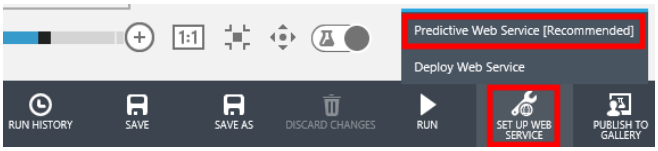
7. Click the **X** in the top right corner of the **Visualize** dialogue box to close it.

Publish a Model as a Web Service

Create a Predictive Experiment from the Training Experiment

Once you are satisfied with the results of a training experiment, you need to convert that training experiment into an experiment that is optimized for predicting new data. For this, you will create a Predictive Experiment.

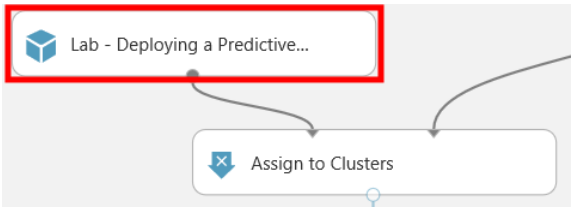
1. Click **SET UP WEB SERVICE** at the bottom of the page and select **Predictive Web Service [Recommended]**.



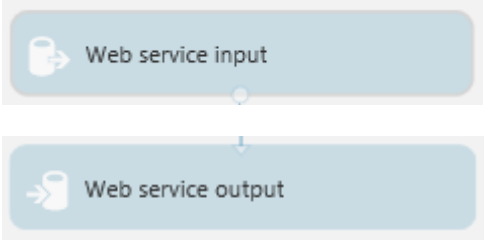
2. Notice a few things:

   a. There are now 2 tabs at the top of the ML Studio canvas: one for the *Predictive experiment*, and one for the *Training experiment*.

   b. The modules for training the model have all been replaced with a new module representing the trained clustering model. The new trained model module feeds the *Assign to Clusters* module.



   c. Modules for *Web service input* and *Web service output* have been added to the predictive experiment. These modules indicate where data would be passed into and out of an experiment that is published as a web service in a production environment.



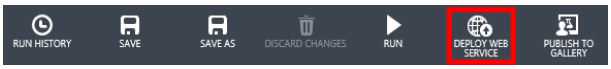
3. Click **RUN** at the bottom of the page to execute the experiment.

   The experiment will execute using the input data set (same data as the Training experiment). It will score all of the rows using the trained clustering model.

Publish the Predictive Experiment as a Web Service

Next, you will publish your predictive experiment as a web service so that predictions can be made from other applications outside of the ML Studio. Once published, you have the ability to create a web or desktop application that can now utilize your predictive model.

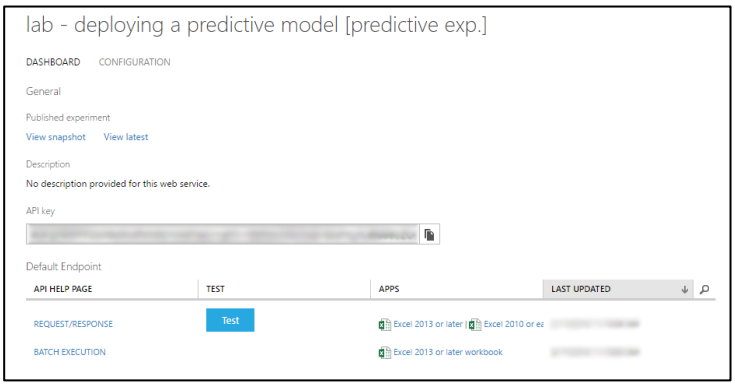
1. Click **DEPLOY WEB SERVICE** at the bottom of the page.



   After you click *DEPLOY WEB SERVICE*, a message box will appear at the bottom of the screen that says *Creating predictive experiment*. After the experiment has been created, click the 

button.

   The web service is created, and you now see the web service dashboard.



   Here you will find:

   a. *Links* to the latest experiment that was used to create the web service

   b. A *description* of the web service

   c. The *API key* used for authentication when other applications call the web service

   d. Other information about the *Default Endpoint* including:

      a. help pages for the *REQUEST/RESPONSE* and *BATCH EXECUTION* APIs

b. a *TEST* option

c. *Excel Workbooks* to download

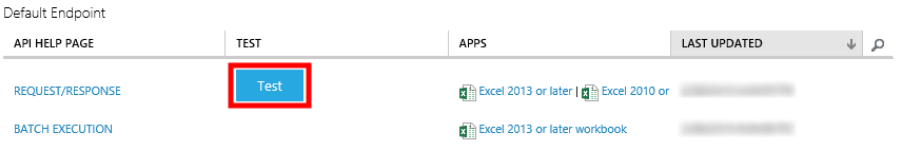
d. data indicating when the endpoint was *LAST UPDATED*

Make Predictions using the Web Service

Test the API using the Web Interface

Finally, we will test our API using the provided Web UI on the web service dashboard page.

1. Click the **Test** button.



2. Enter the following values in their corresponding text boxes on the **Enter data to predict** dialogue box. You may need to scroll down to view all fields depending on the screen resolution or the browser window height.

   LOCATION: **Houston, TX**

   GINIINDEX: **0.4802**

   POP2014: **2239558**

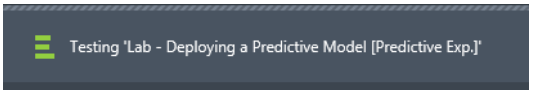
   LANDAREA: **599.6**

   POPULTREND: **6.63**

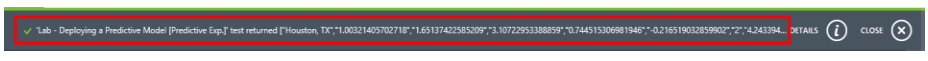
   POPDENSITY: **3735**

3. Click the  at the bottom of the **Enter data to predict** dialogue box.

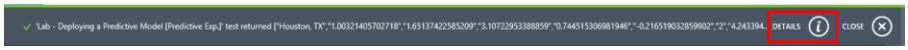
   Notice a status bar at the bottom of the page is displayed indicating the web service call is being made.



   Once the web service call is complete, the status bar will have a *green checkmark* next to the web service name along with the returned prediction from the API.



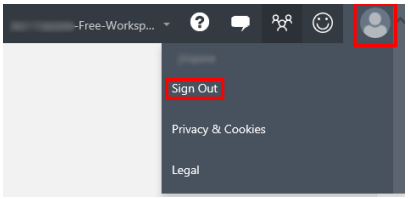
4. Click **DETAILS** on the status bar.



The status bar will expand to show the entire result body that was returned from the API including the cluster number. Based on the characteristics of the input data, the clustering model assigned the observation to an existing cluster as output.

5. Click **CLOSE** on the status bar to return to the web service dashboard.

6. Sign out of your workspace by clicking the **profile picture** at the top right of the page and selecting **Sign Out** from the displayed menu.



Conclusion

This concludes the *Deploying a Predictive Model with Azure Machine Learning* lab. To recap, you have successfully copied an experiment from the Cortana Intelligence Gallery into ML Studio, created a predictive experiment, deployed a model as a web service, and interacted with the API via the included web interface.

Now that you have a deployed web service, you can use the provided web UI or the provided Excel Worksheet to predict with new data. You can also create custom applications and data flows that access the API for automated prediction.

To answer our earlier question of where to open your next restaurant location, you would start by downloading the cluster assignments into a tool like Excel to try to label the type of city-market each cluster represents. Then you could take those cluster labels and compare them to your restaurant locations that have done well in the past. This would give you a list of prime city-markets for future restaurant locations.

Terms of Use

By using this Hands-on Lab, you agree to the following terms:

The technology/functionality described in this Hands-on Lab is provided by Microsoft Corporation in a "sandbox" testing environment for purposes of obtaining your feedback and to provide you with a learning experience. You may only use the Hands-on Lab to evaluate such technology features and functionality and provide feedback to Microsoft. You may not use it for any other purpose. You may not modify, copy, distribute, transmit, display, perform, reproduce, publish, license, create derivative works from, transfer, or sell this Hands-on Lab or any portion thereof.

COPYING OR REPRODUCTION OF THE HANDS-ON LAB (OR ANY PORTION OF IT) TO ANY OTHER SERVER OR LOCATION FOR FURTHER REPRODUCTION OR REDISTRIBUTION IS EXPRESSLY PROHIBITED.

THIS HANDS-ON LAB PROVIDES CERTAIN SOFTWARE TECHNOLOGY/PRODUCT FEATURES AND FUNCTIONALITY, INCLUDING POTENTIAL NEW FEATURES AND CONCEPTS, IN A SIMULATED ENVIRONMENT WITHOUT COMPLEX SET-UP OR INSTALLATION FOR THE PURPOSE DESCRIBED ABOVE. THE TECHNOLOGY/CONCEPTS REPRESENTED IN THIS HANDS-ON LAB MAY NOT REPRESENT FULL FEATURE FUNCTIONALITY AND MAY NOT WORK THE WAY A FINAL VERSION MAY WORK. WE ALSO MAY NOT RELEASE A FINAL VERSION OF SUCH FEATURES OR CONCEPTS. YOUR EXPERIENCE WITH USING SUCH FEATURES AND FUNCITONALITY IN A PHYSICAL ENVIRONMENT MAY ALSO BE DIFFERENT.

FEEDBACK. If you give feedback about the technology features, functionality and/or concepts described in this Hands-on Lab to Microsoft, you give to Microsoft, without charge, the right to use, share and commercialize your feedback in any way and for any purpose. You also give to third parties, without charge, any patent rights needed for their products, technologies and services to use or interface with any specific parts of a Microsoft software or service that includes the feedback. You will not give feedback that is subject to a license that requires Microsoft to license its software or documentation to third parties because we include your feedback in them. These rights survive this agreement.

MICROSOFT CORPORATION HEREBY DISCLAIMS ALL WARRANTIES AND CONDITIONS WITH REGARD TO THE HANDS-ON LAB , INCLUDING ALL WARRANTIES AND CONDITIONS OF MERCHANTABILITY, WHETHER EXPRESS, IMPLIED OR STATUTORY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. MICROSOFT DOES NOT MAKE ANY ASSURANCES OR REPRESENTATIONS

WITH REGARD TO THE ACCURACY OF THE RESULTS, OUTPUT THAT DERIVES FROM USE OF THE VIRTUAL LAB, OR SUITABILITY OF THE INFORMATION CONTAINED IN THE VIRTUAL LAB FOR ANY PURPOSE.

DISCLAIMER

This lab contains only a portion of the features and enhancements in Microsoft Azure Machine Learning. Some of the features might change in future releases of the product.