Data Science and Machine Learning Essentials

Lab 5B – Publishing Models in Azure ML

By Stephen Elston and Graeme Malcolm

Overview

In this lab you will publish one of the models you created in a previous lab as a web service. You will then test the web service by interacting with the web service through an Excel plugin.

Note: This lab builds on the experiment you completed in Lab 4A, Regression. If you have not completed that lab, you can copy the experiment from the Cortana Analytics Gallery.

What You'll Need

To complete this lab, you will need the following:

- An Azure ML account.
- A web browser and Internet connection.

Note: To set up the required environment for the lab, follow the instructions in the **Setup** document for this course.

Preparing an Experiment for Publishing

In this lab, you will create a web service based on the energy efficiency regression experiment you created in a previous lab. To be publishable as a web service an experiment should have a single machine learning model. You remove some of the experimental models you created in your regression experiment.

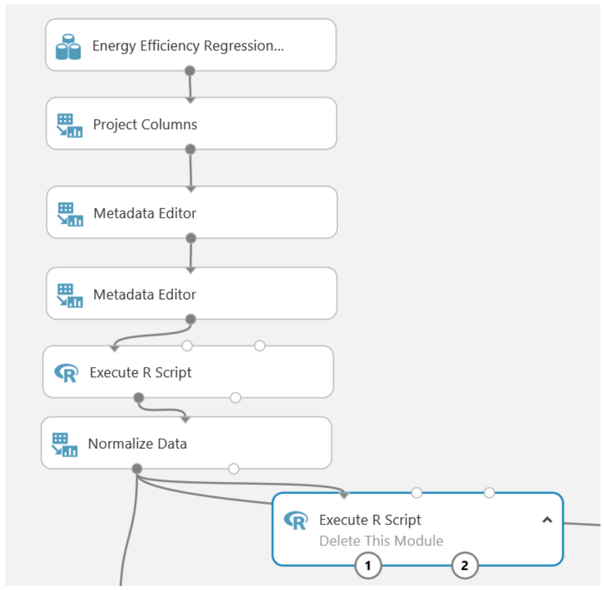
Create a Copy of the Regression Experiment

1. If you have not already done so, open a browser and browse to https://studio.azureml.net. Then sign in using the Microsoft account associated with your Azure ML account.
2. If you prefer to work with R, open the **Regression (R)** experiment you created for Lab 4A, and save a copy as **Web Service (R)**. If you prefer to work with Python, save a copy of the **Regression (Python)** experiment as **Web Service (Python)**.
   If you did not complete Lab 4A, you can copy the appropriate **Regression** experiment from the collection for this course in the Cortana Analytics Gallery at http://gallery.cortanaanalytics.com/Collection/5bfa7c8023724a29a41a4098d3fc3df9.

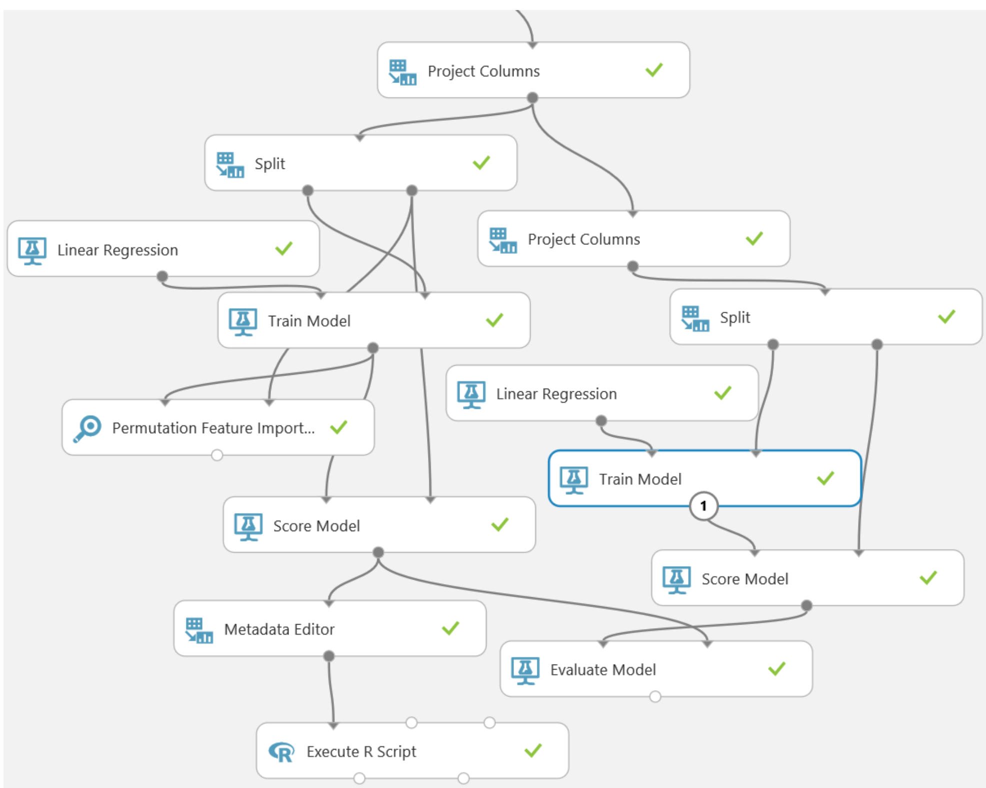
Prepare Your R Experiment

Note: If you prefer to work with Python, skip this procedure and complete the *Prepare Your Python Experiment* procedure that follows it.

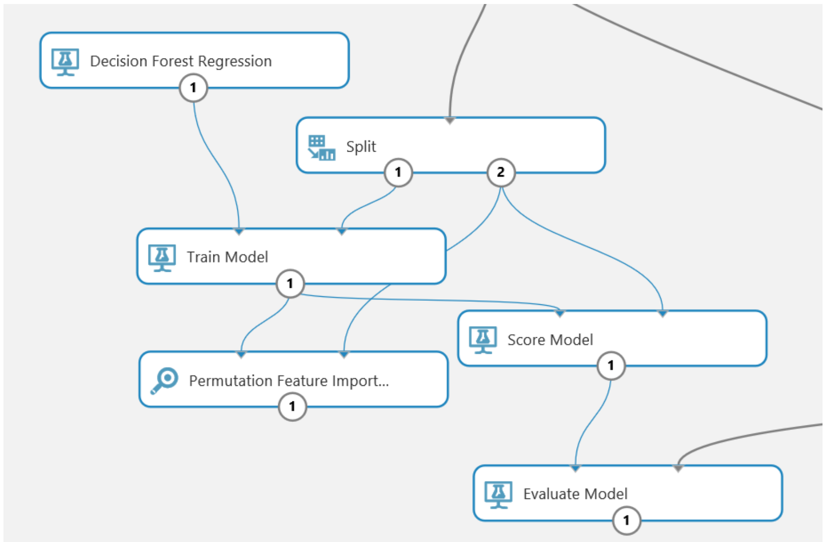
1. In the **Web Service (R)** experiment, in the first section of the data flow (before the models are defined), delete the **Execute R Script** module that is connected to the output of the **Normalize Data** module and used to visualize data (shown below):



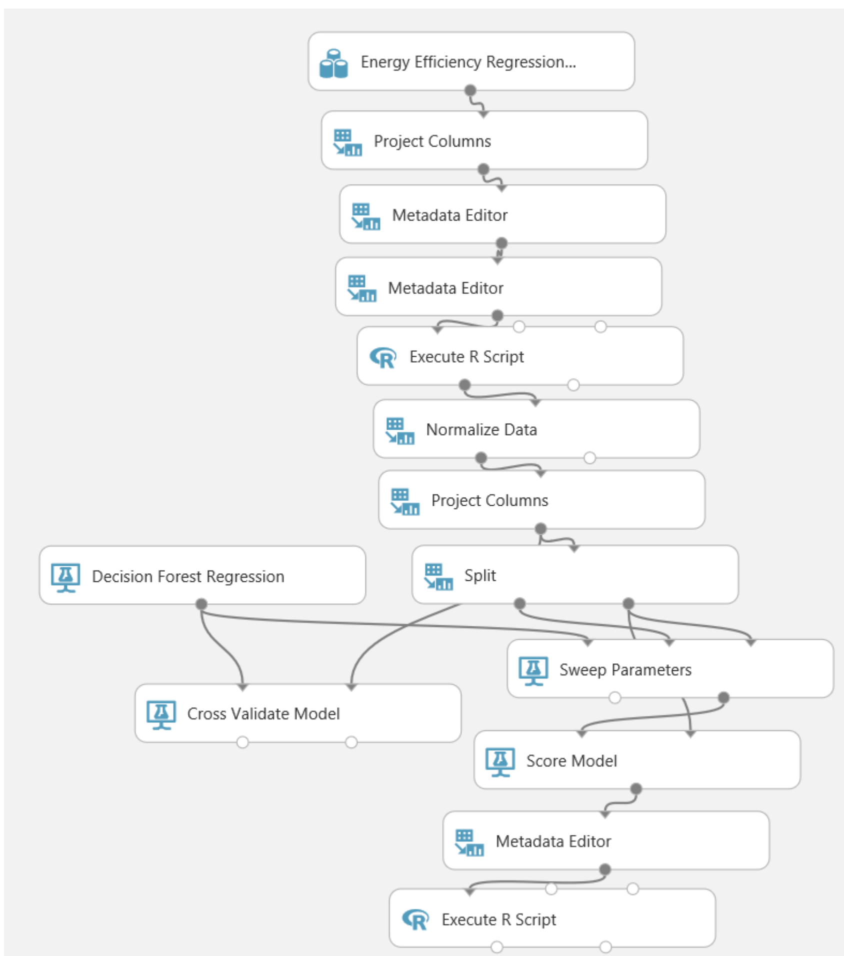
2. Delete the portion of the experiment containing the two **Linear Regression** model modules (shown below), leaving only the two **Decision Forest Regression** models.



3. Delete the portion of the experiment containing the **Decision Forest Regression**, **Split**, **Train Model**, **Permutation Feature Importance**, **Score Model**, and **Evaluate Model** modules and shown in the figure below:



4. Verify that the experiment now consists only of the initial data preparation flow, and the **Decision Forest Regression** model that uses the **Sweep Parameters** module. It should resemble the following image.

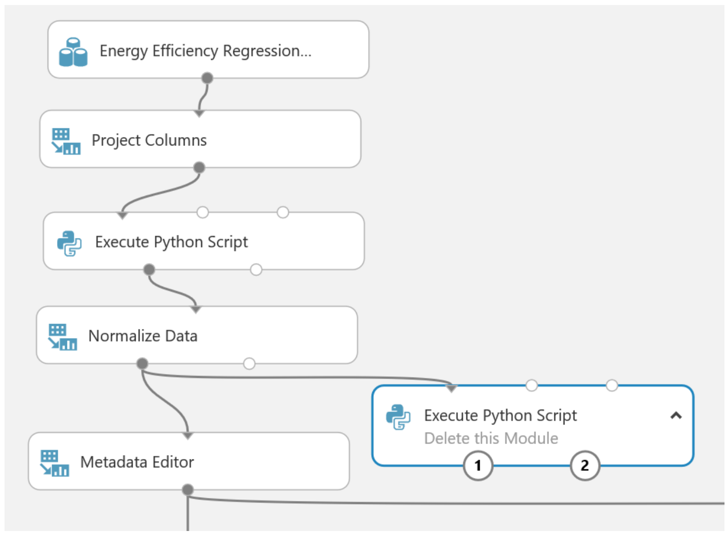


5. Save your experiment.

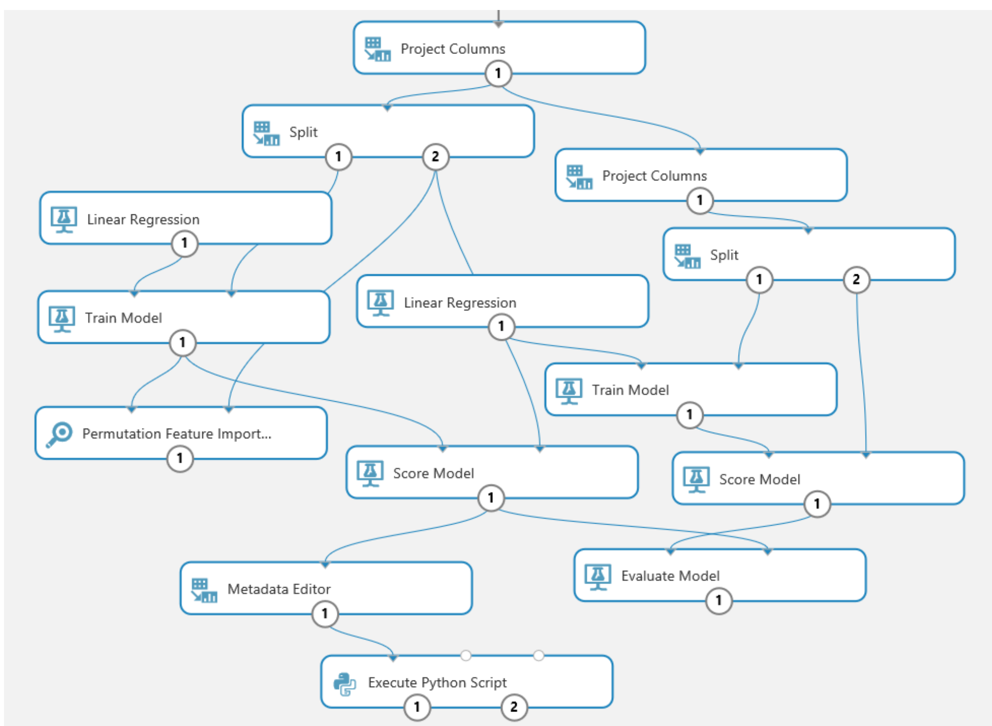
Prepare Your Python Experiment

Note: If you prefer to work with R, skip this procedure and complete the preceding *Prepare Your R Experiment* procedure.

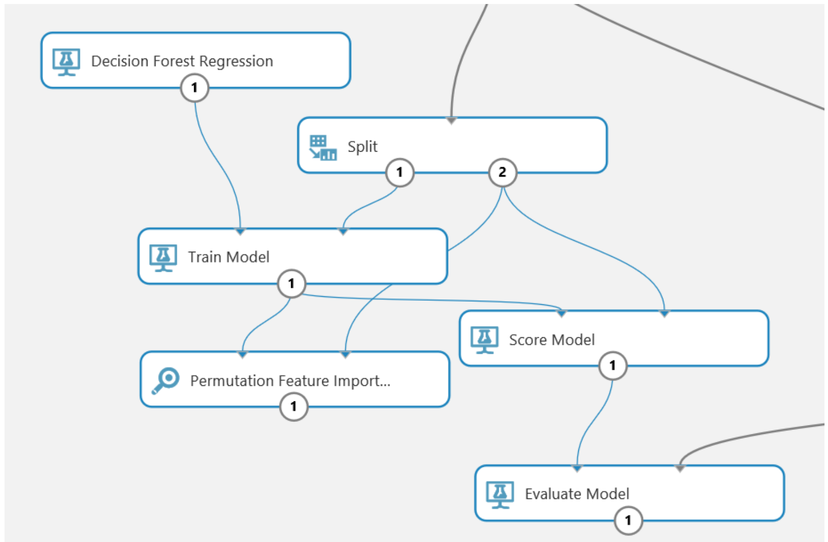
1. In the **Web Service (Python)** experiment, in the first section of the data flow (before the models are defined), delete the **Execute Python Script** module that is connected to the output of the **Normalize Data** module and used to visualize data (shown below):



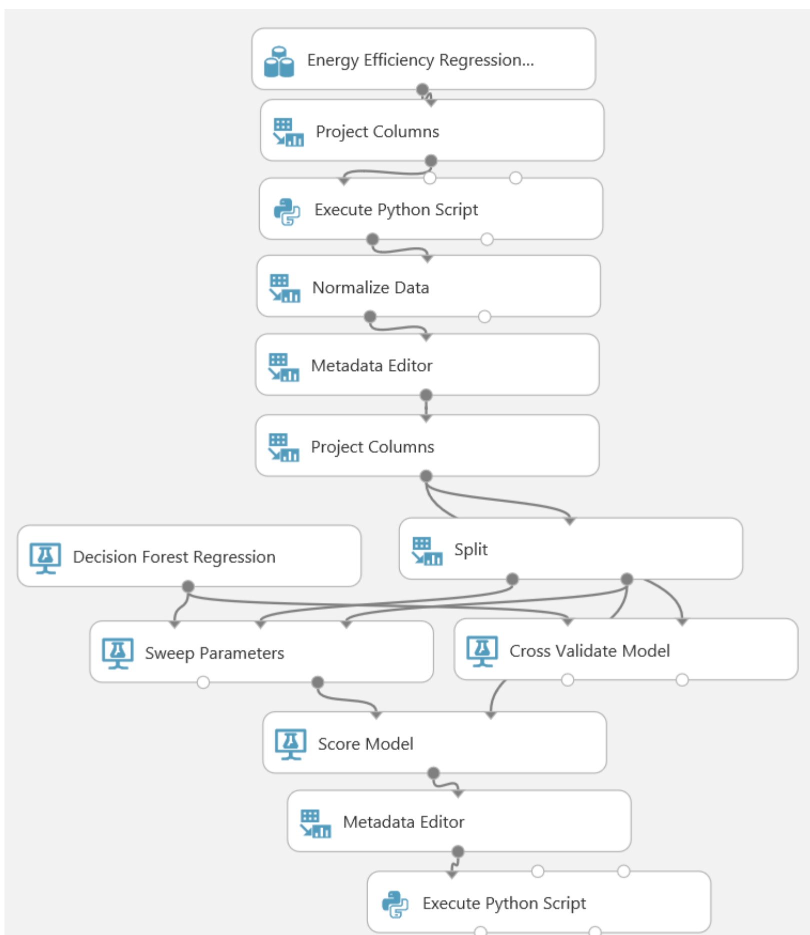
2. Delete the portion of the experiment containing the two **Linear Regression** model modules (shown below), leaving only the two **Decision Forest Regression** models.



3. Delete the portion of the experiment containing the **Decision Forest Regression**, **Split**, **Train Model**, **Permutation Feature Importance**, **Score Model**, and **Evaluate Model** modules and shown in the figure below:



4. **Delete** these modules.
5. Ensure that your experiment resembles the following figure:



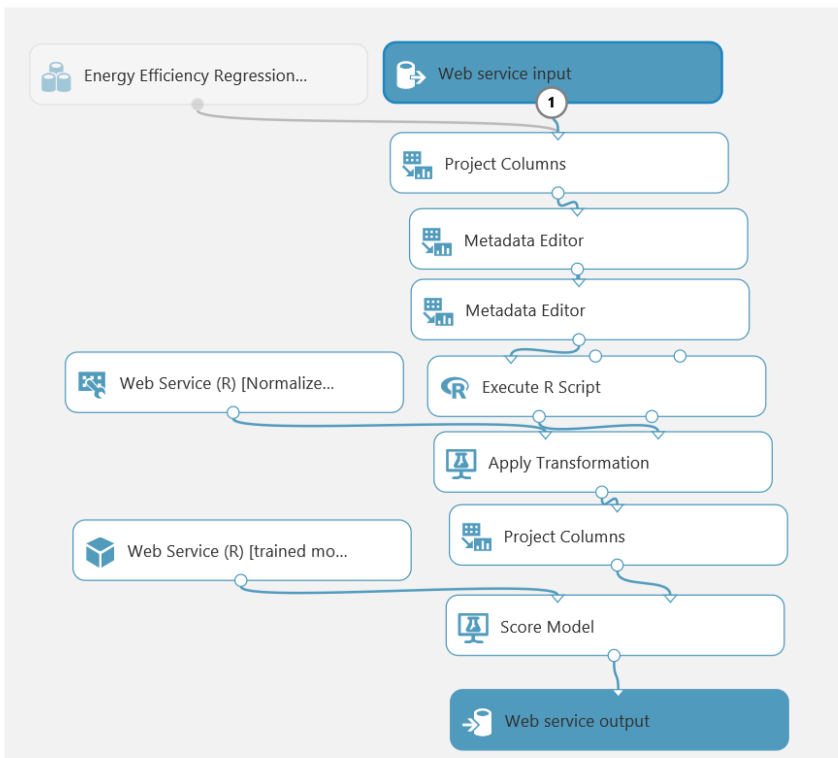
6. Save your experiment.

Publishing a Model

You have prepared your experiment (starting with either the R or Python version) by pruning the experiment to contain just the one best decision forest model. You are now ready to create and a **Predictive experiment**.

Create a Web Service

1. Run the experiment and make sure there are no errors.
2. At the bottom of the page, click **Set Up Web Service**, and select **Predictive Web Service [Recommended]**. A predictive experiment is created (the R version is shown below.)



Notice the following aspects of the predictive experiment:

- A **Web service input** module has been added alongside the original dataset.
- The original **Energy Efficiency Regression** dataset has been retained. This dataset defines the input schema for the web service and is used for testing. Do not remove this dataset.
- A series of modules which perform row-at-a-time transformations, **Project Columns**, two **Metadata Editors**, and **Execute R Script** (or **Execute Python Script**) modules, remain as before.
- The original **Normalize Data** module has been converted into an **Apply Transformation** module and the **Web Service (R) [Normalize Data]** (or **Web Service (Python) [Normalize Data])** module which defines the transformation created from the training data.

- The second **Project Columns** module remains unchanged.
- The **Decision Forest Regression** module has been transformed into a **Web Service (R) [trained model]** (or **Web Service (Python) [trained model]**) module, which contains the trained model.
- The **Score Model** module remains.
- A **Web services output** module has been added to the **Predictive experiment**.

3. Save and Run the predictive experiment.
4. To ensure the experiment ran correctly visualize the output of the **Score Model** module and check that your results resemble the following:

rows	columns
768	7

view as

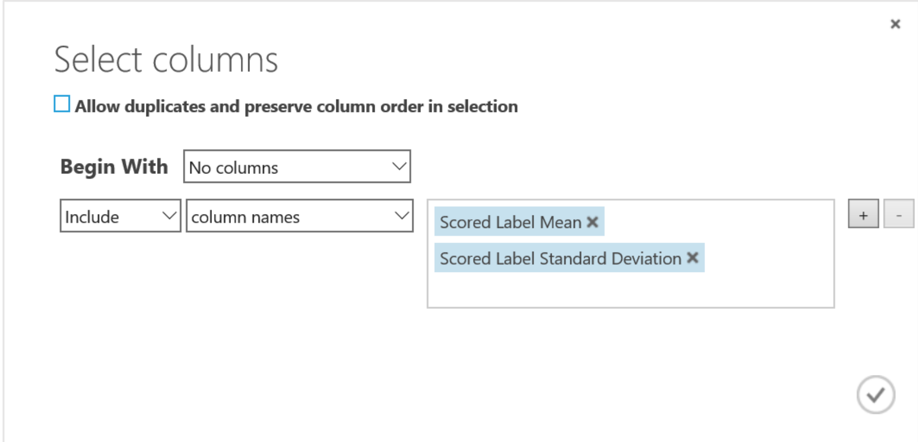
WallArea	RoofArea	OverallHeight	GlazingArea	HeatingLoad	Scored Label Mean	Scored Label Standard Deviation
0.285714	0	7	0	15.55	15.55	0.000183
0.285714	0	7	0	15.55	15.55	0.000183
0.285714	0	7	0	15.55	15.55	0.000183
0.285714	0	7	0	15.55	15.55	0.000183
0.428571	0.111111	7	0	20.84	20.84	0.000224
0.428571	0.111111	7	0	21.46	20.84	0.000224
0.428571	0.111111	7	0	20.71	20.84	0.000224
0.428571	0.111111	7	0	19.68	20.84	0.000224
0.285714	0.333333	7	0	19.5	17.136667	0.241937
0.285714	0.333333	7	0	19.95	17.136667	0.241937
0.285714	0.333333	7	0	19.34	17.136667	0.241937
0.285714	0.333333	7	0	18.31	17.136667	0.241937

Note that the web service returns all of the columns in the output of the **Score Model** module, including the feature columns that will be supplied by the client application calling the web service. You will remove this data redundancy in the next procedure.

5. Close the visualization.

Customize the Web Service Output

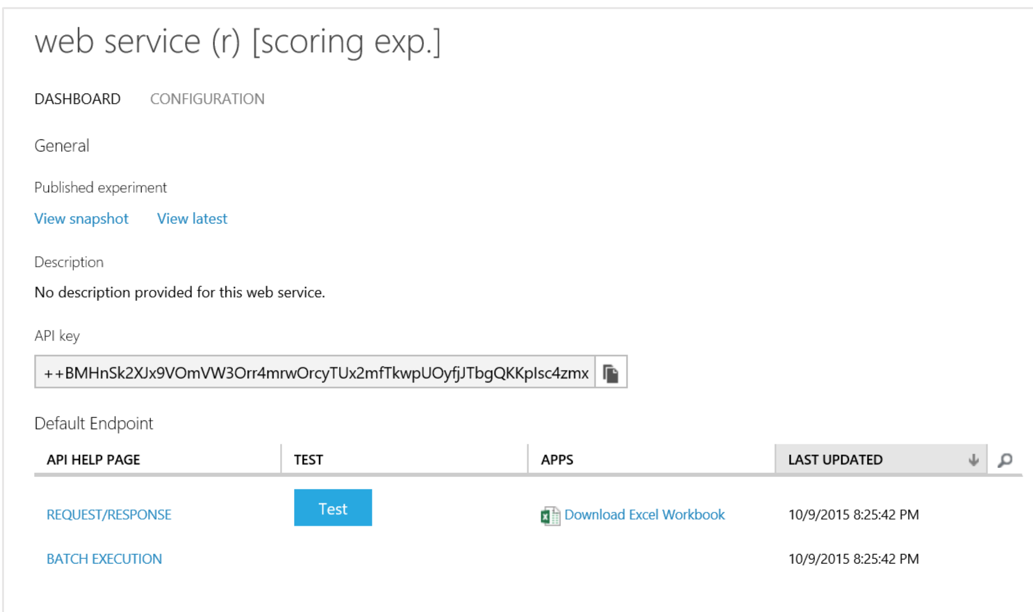
1. Delete the connection between the **Score Model** module and the **Web service output**.
2. Search for a **Project Columns** module and drag it onto the canvas between the **Score Model** module and the **Web service output**. Then connect the output from the **Score Model** module to the input of the **Project Columns** module, and connect the output of the **Project Columns** module to the input of the **Web service output**.
3. Select the **Project Columns** module you just added, and launch the column selector to select only the **Scored Label Mean** and **Scored Label Standard Deviation** columns as shown here:



4. Save and run the predictive experiment again, and visualize the output of the **Project Columns** module to verify that only two columns are returned. Then close the visualization.

Deploy the Web Service

1. Click the **Deploy Web Service** icon at the bottom of the page. After the process has finished, you should see the web service properties page as shown below (if you started with a Python experiment the title line will read, **Web Service (Python)[Scoring Experiment]**):

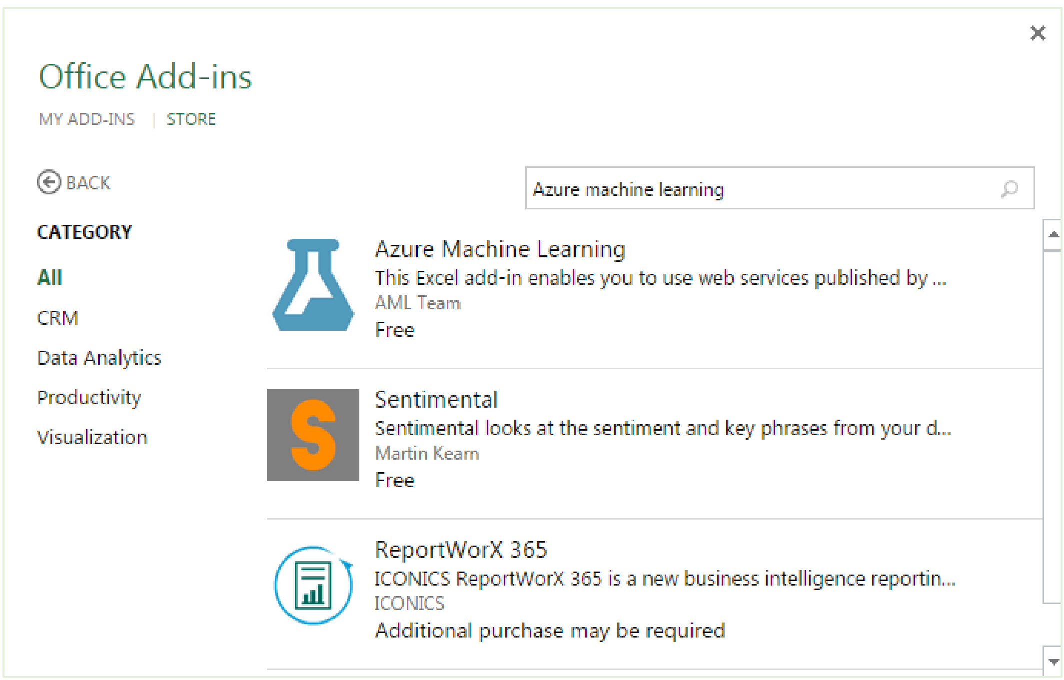


2. Open a text editor such as Notepad. Then click the **Copy** icon next to the **API key** to copy it to the clipboard, and paste it into the text editor – you will need this later to test the web service.
3. Click the **Request Response** hyperlink on your web services properties page. This opens a new browser tab containing the Request Response API documentation for your web service with a URL starting with *https://studio.azureml.net/apihelp/*.
4. Copy the URL in the address bar for the *Request Response API Documentation* page that opened and paste it into the text editor – you will also need this to test the web service.

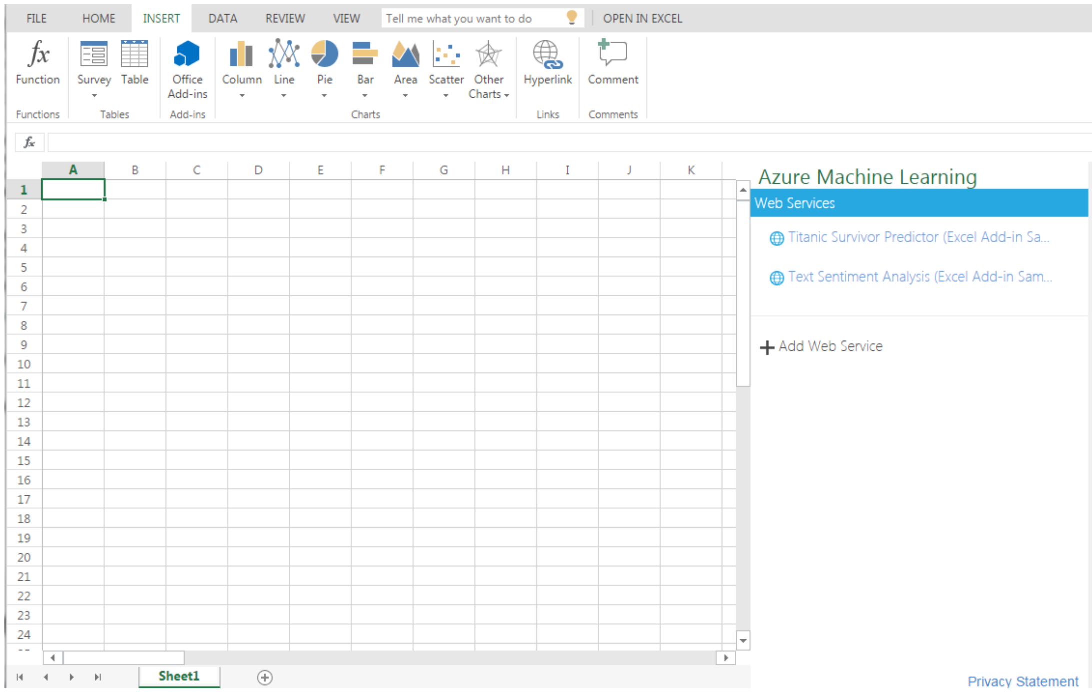
Test the Web Service

In this procedure you will test your web service using Excel with a plugin and the dataset you downloaded at the beginning of this lab.

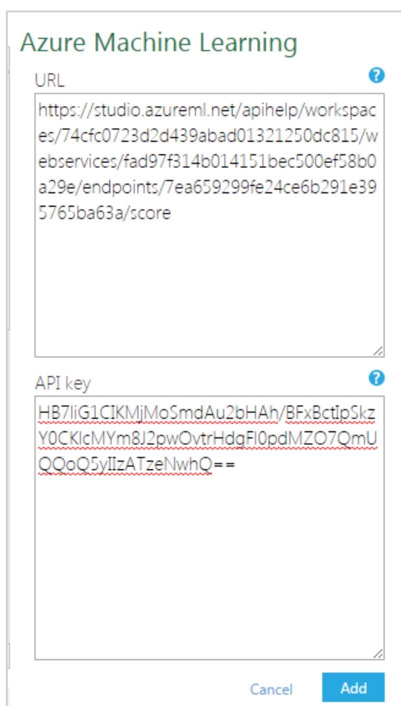
1. If you do not already have a **OneDrive** account navigate to [https://**onedrive**.live.com/](https://onedrive.live.com/) and follow the directions to establish an account using the same Microsoft account you use to access **Azure ML Studio**. You can now access **Azure ML Studio** and **OneDrive** using the same account.
2. In the **New** menu, click **Excel workbook**. A new browser tab with the workbook will appear.
3. On the **File** tab, click **Save as** and then save the workbook as **Web Service Test** in your OneDrive.
4. On the **Insert** tab, click **Office Add-ins**.
5. In the **Office Add-ins** dialog box, click **Store**. Then type **Azure Machine Learning** in the search box and click the **search icon** (magnifying glass), and verify that you see a dialog box resembling the following:



6. Click the **Azure Machine Learning** icon, and on the popup dialog click **Trust it**.
7. After the Azure Machine Learning plugin has loaded verify that your workbook resembles the following image:

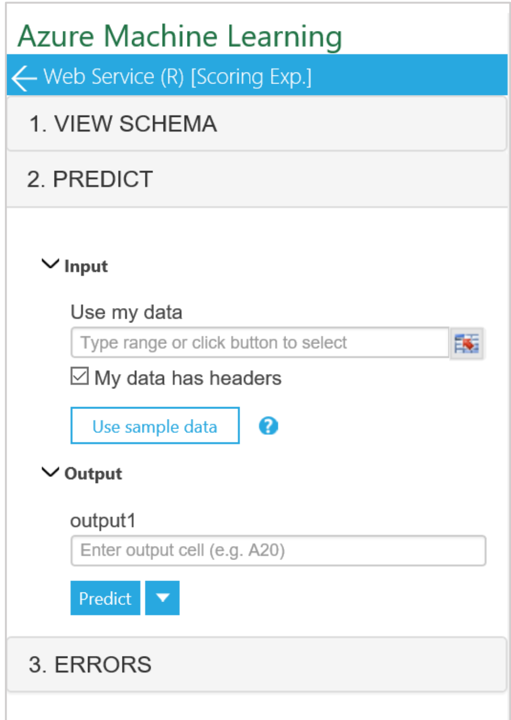


8. In the Azure Machine Learning pane, click **+ Add Web Service**.
9. Paste the Request Response URL you copied to the text editor earlier into the **URL** box of the **Azure Machine Learning plugin** in your workbook.
10. Paste this API key you copied to the text editor earlier into the **API key** box of the **Azure Machine Learning plugin** in your workbook.
11. Verify that the **Azure Machine Learning plugin** in your workbook resembles the figure below:



12. Click the **Add** button.

13. After the web service has been added to your workbook verify that the **Azure Machine Learning plugin** resembles the following figure (note if you started with a Python experiment the title will read **Web Service (Python)[Scoring Exp]**):



14. Notice that the **API key** and **URL** are no longer accessible. The **API key** is now encrypted making the workbook safely redistributable.
15. Leaving the **My data has headers** box checked, click the **Use sample data** button.
16. Enter the following values into the workbook in cells A2 to J2:

Relative Compactness	Surface Area	Wall Area	Roof Area	Overall Height	Orientation	Glazing Area	Glazing Distribution	Heating Load	Cooling Load
0.715	710	270	220	3.5	2	0.1	2	0	0

17. Select the header and data cells (A1 to J2). Then, in the Azure Machine Learning pane, click the grid icon to the right of the box under **Use my data** and verify that the selection is **'Sheet1'!A1:J2**, and click **OK**.
18. In the box under **output1** type **K1** (a header will be created for the web services response).
19. Click the **Predict** button.
20. When the web service has finished running, view the data returned in cells K2 (Scored Label Mean) and L2 (Scored Label Standard Deviation). These are the two output columns from the web service, and the **Scored Label Mean** column contains the predicted value for heating load.
21. Congratulate yourself! You have successfully created a machine learning web service.

Summary

In this lab you have published an Azure ML model as a web service. Specifically, you:

- Pruned the models in an experiment to prepare it for publication.
- Created an Azure ML web service.

- Tested the web service in an Excel notebook using the Azure Machine Learning plugin. The workbook you have created is redistributable.

Note: The experiment created in this lab is available in the Cortana Analytics library at http://gallery.cortanaanalytics.com/Collection/5bfa7c8023724a29a41a4098d3fc3df9.