Big Data with Azure Machine Learning

Lab 3 – Publishing Predictive Web Services

Overview

In this lab, you will learn how to publish predictive models as Azure Machine Learning Web Services, and consume them from client applications.

What You'll Need

To complete this lab, you will need the following:

- A Microsoft account (for example, an *outlook.com*, *live.com*, or *hotmail.com* address)
- A Microsoft Azure subscription
- A Windows, Linux, or Mac OS X computer
- The lab files for this course

Note: Before starting this lab, you must complete the previous labs in this course.

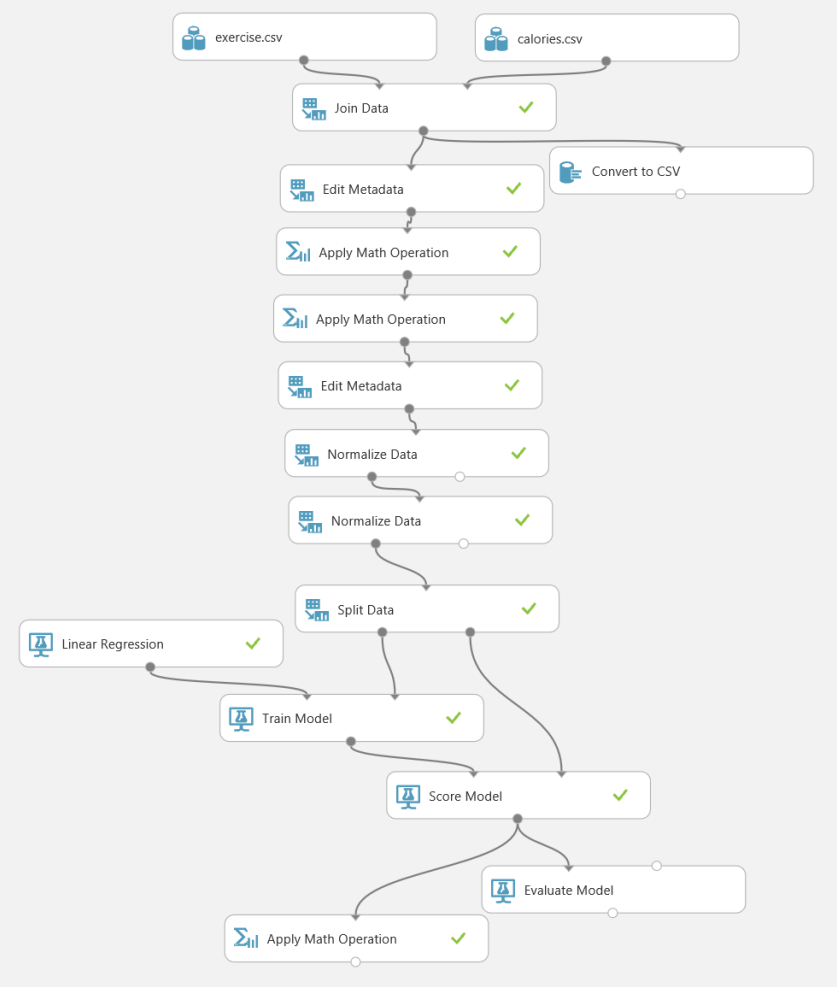
Exercise 1: Publishing and Testing a Web Service

In this exercise, you will publish and test the calorie expenditure regression model you created in previous labs as a web service.

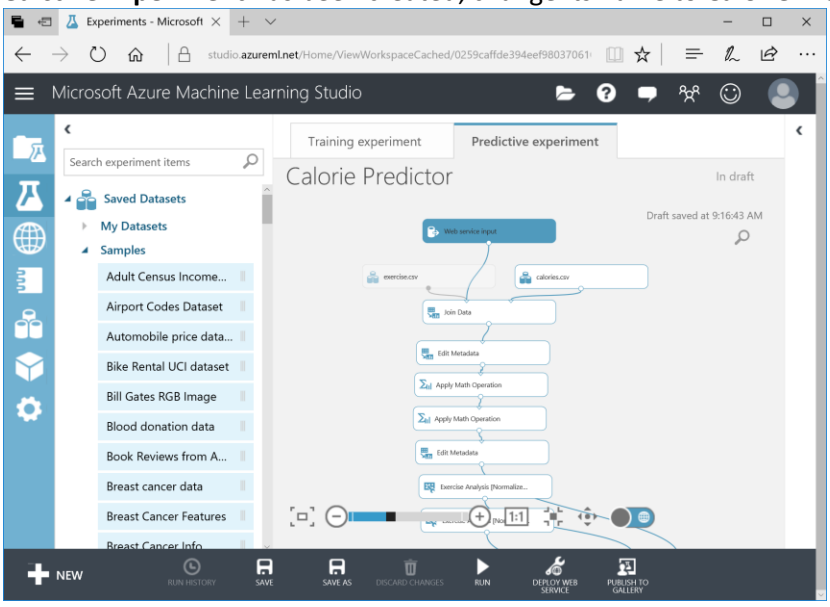
Create a Predictive Experiment

In the previous lab, you trained and evaluated a regression machine learning model that predicts calories expended based on physiological and exercise features. You will now create a predictive experiment for this model in preparation for deploying a web service.

1. If you are not already signed into Azure Machine Learning Studio, browse to https://studio.azureml.net and sign in using the Microsoft account associated with your Azure subscription.
2. Open the **Exercise Analysis** experiment you created in the previous labs, which should look like this:

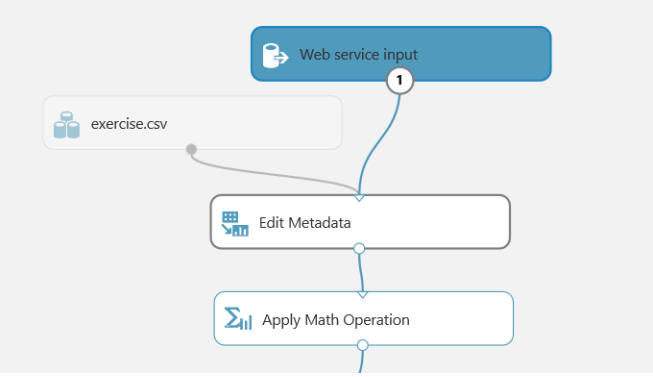


3. Run the experiment and ensure there are no errors.
4. On the **Set Up Web Service** menu, click **Predictive Web Service [Recommended]**. Azure Machine Learning creates a new tab in Azure Machine Learning Studio and generates a predictive experiment from your training experiment.
5. After the **Predictive Experiment** has been created, change its name to **Calorie Predictor**:



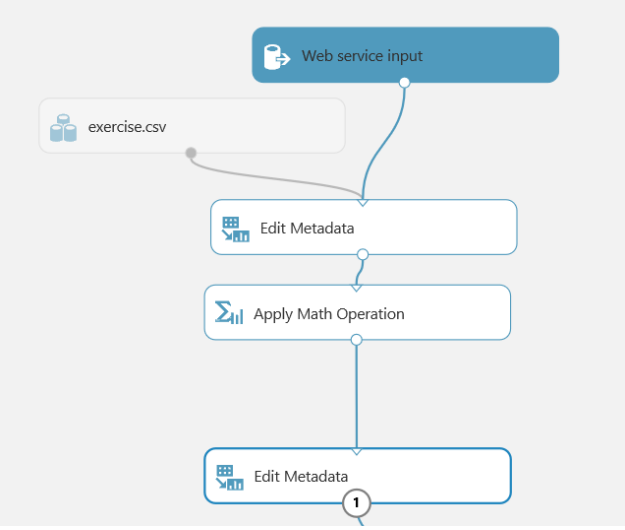
Note: The predictive experiment defines a web service input (named **input1** by default) based on the training dataset in your training experiment. However, your training experiment also includes a dataset that contains the label that the web service will predict (**Calories**). Since this is what the trained model in the web service will predict, it is unnecessary to include the training labels in the predictive experiment.

6. Delete the **calories.csv** data set and the **Join** module. Then connect the **exercise.csv** dataset and the **Web Services Input** to the input of the first **Edit Metadata** module, as shown here:



Note: Now that you have removed the **Calories** column from the data flow, you must remove any explicit references to this column in the rest of the experiment.

7. Delete the second **Apply Math Operation** module, which computes the log of **Calories**; and connect the first (and now only) **Apply Math Operation** module to the second **Edit Metadata** module, as shown here:



8. Change the properties of the second **Edit Metadata** module to eliminate the reference to the **Calories** column as follows.
    - **Column**: User_ID
    - **Data type**: Unchanged
    - **Categorical**: Unchanged
    - **Fields**: Clear feature
    - **New column names**: *Leave blank*

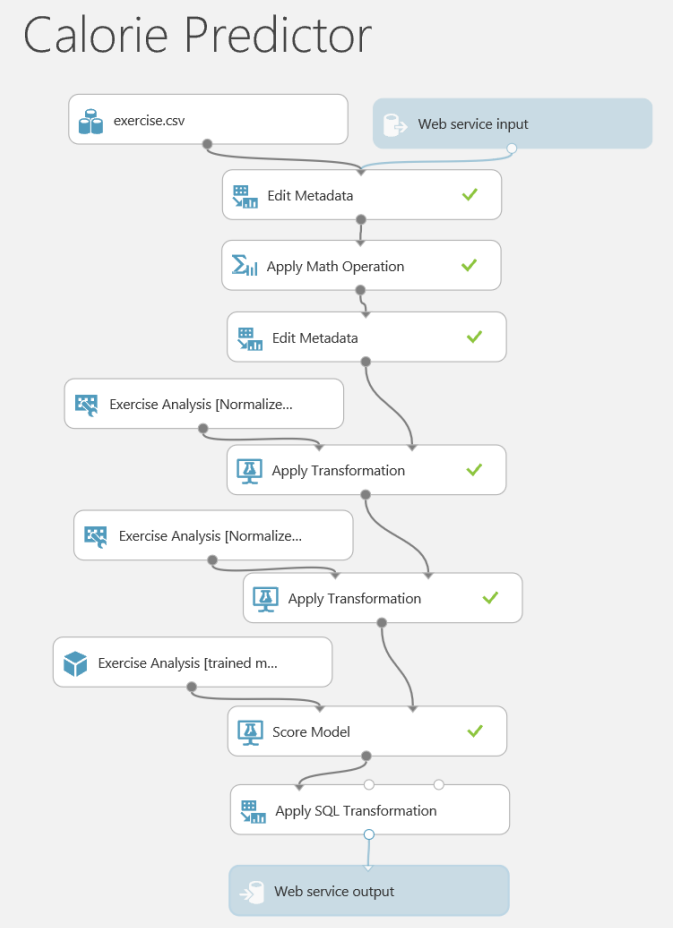
9. Examine the rest of the experiment. It contains two transformations that encapsulate the normalization calculations generated from the training data statistics, and a trained model based on the training algorithm and configuration in the training experiment.
10. Save and run the experiment to verify that no errors occur as a result of removing the **Calories** column.
11. When the experiment has finished running, visualize the output of the **Score Model** module, which is connected to the **Web service output** and therefore reflects the results that will be returned by the web service.

    Note: The output includes all the columns in the data flow, many of which are normalized versions of the input fields. Additionally, the **Scored Labels** column, which contains the value predicted by the model, reflects the log-normal of **Calories** – not the actual number of calories.

12. Delete the connection between the **Score Model** module and the **Web Services Output**.
13. Add an **Apply SQL Transformation** module to the experiment and connect the output of the **Score Model** module to its **Table1** (leftmost) input.
14. Configure the **Apply SQL Transformation** module properties to update the SQL query script to the following code (which you can copy and paste from **Calories SQL.txt** in the **Lab03** folder):

```
SELECT exp([Scored Labels]) AS Calories FROM t1;
```

15. Connect the output of the **Apply SQL Transformation** module to the input of the **Web service output**.
16. Verify that, with a bit of rearranging, your experiment resembles the figure below:



17. Save and run the experiment.
18. Visualize the output of the **Apply SQL Transformation** module, and verify that the predicted **Calories** values are returned when the experiment is run with the test data.

Deploy the Web Service

Now that you have prepared the predictive experiment, you can deploy it as a web service.

Note: Azure Machine Learning supports two kinds of web service: "classic" and "new". *Classic* web services are managed separately from an Azure subscription while *new* web services are compatible with the Azure resource management model, and are managed as resources in resource groups within an Azure subscription. In this course, you will work with new web services. Note that the free-tier Azure Machine Learning workspace supports only classic web services (as it is not associated with pricing plan in an Azure subscription). Client applications consume classic and new web services in the same way, but there are some differences in the way they are managed.

1. With the **Calorie Predictor** predictive experiment open, in the **Deploy Web Services** menu, click **Deploy Web Service [New] Preview**. A web page with a header **Deploy "Calorie Predictor" experiment as a web service** will appear.
2. Set the web service configuration as follows:
    - **Web Service Name**: CaloriePredictor
    - **Storage Account**: The Azure storage account associated with your workspace
    - **Pricing Plan**: The Azure Machine Learning pricing plan you created with your workspace.
3. Click the **Deploy** box in the lower left of the page. After some time, the **Quickstart** page for your **Calorie Predictor** web service should appear.

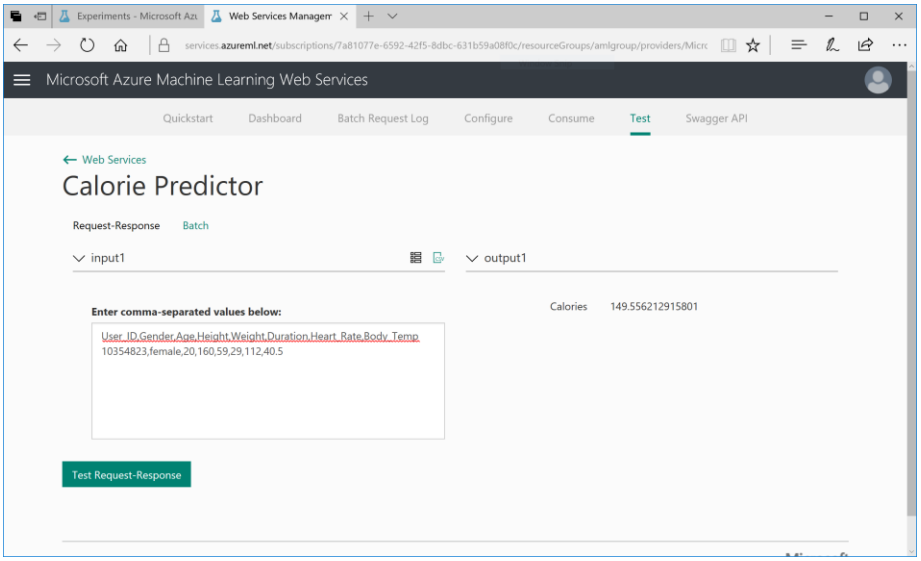
Test the Web Service

Now that you have deployed your predictive web service, you should test it.

1. In the **Quickstart** page for your web service, click **Test**.
2. With the **Request-Response** endpoint selected, for **input1**, click the **CSV** icon. Then replace the default test input fields with the following (which you can copy and paste from the **RequestResponse-Test.txt** file in the **Lab03** folder):

```
User_ID,Gender,Age,Height,Weight,Duration,Heart_Rate,Body_Temp
10354823,female,20,160,59,29,112,40.5
```

3. Click **Test Request-Response** and view the output that is returned, as shown here:



4. Click **Batch** to test the batch endpoint for your web service.
5. Under **input1**, browse to the **Lab03** folder and select **Batch-Test.csv**.
6. Click **Test**, and wait for the batch job status to change to **Finished**. This may take a minute or so. (If the status does not change, try viewing the **Batch Request Log** page and refreshing the page.)
7. Under **result**, click **output1** and open the response returned by the batch operation in your preferred application for CSV documents.

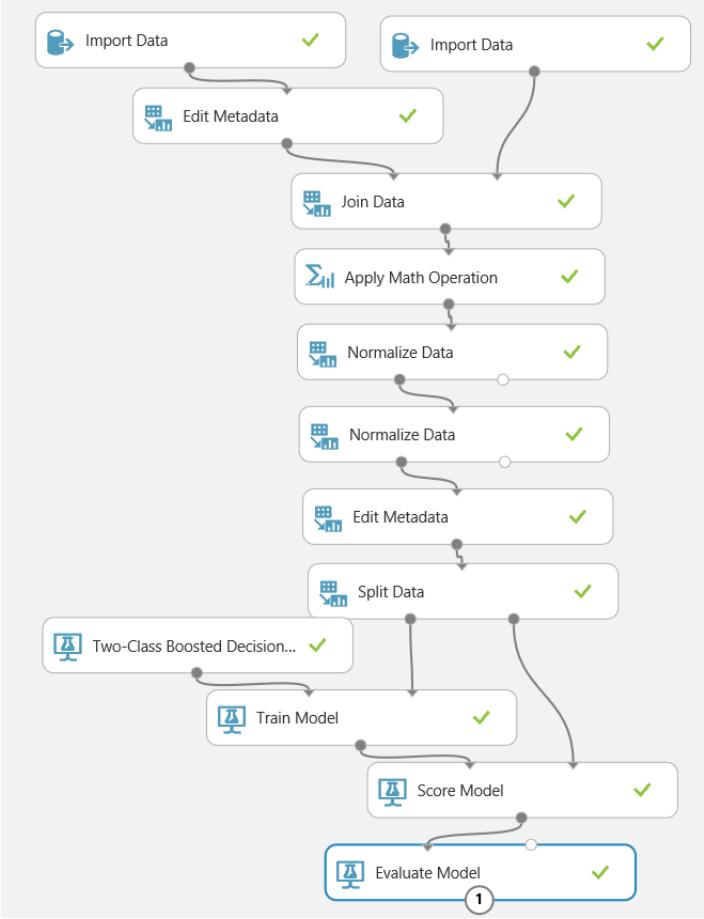
Exercise 2: Publishing and Consuming a Parameterized Web Service

In this exercise, you will publish and consume the diabetes classification model you created in previous labs as a parametrized web service.

Create a Predictive Experiment

Now that you have created and evaluated a machine learning you will publish the model as a web service.

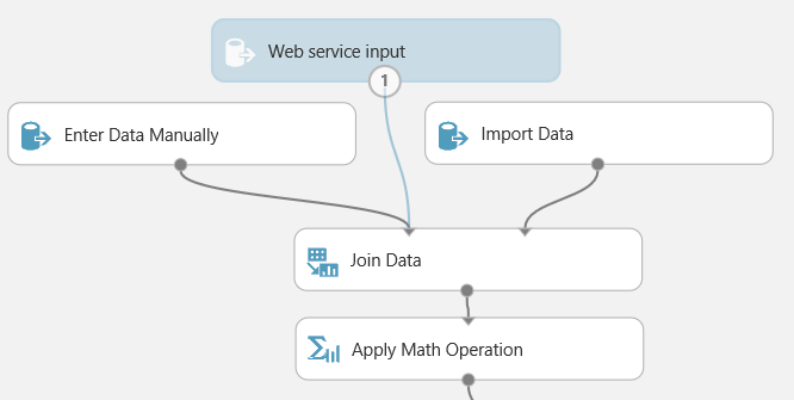
1. In Azure Machine Learning Studio, open the **Pima Big Data** experiment you created in the previous labs, which should look like this:



2. Run the experiment.
3. On the **Set Up Web Service** icon at the bottom of the studio screen, click **Predictive Web Service [Recommended]**. A new tab containing a **Predictive Experiment** will appear.
4. Change the name of the predictive experiment to **Diabetes Predictor**.

Note: The **input1** web service input in the predictive experiment is based on the training dataset in your training experiment, which is imported from text files in Azure storage. These files include the label (**Diabetes**), which is what the web service will predict – so it does not seem sensible to require this as an input parameter. Additionally, the files do not include column headers, so the input schema for the web service will consist of parameters named **Col1**, **Col2**, and so on. You will address these problems by defining a custom input for the predictive experiment.

5. Delete the **Import Data** module that imports the patient data from Azure Storage (not the one that imports physician data from Azure SQL Database). Then delete the **Edit Metadata** module that it was connected to (which changes the column names).
6. Add an **Enter Data Manually** module to the experiment and connect its output to the left input of the **Join Data** module.
7. Connect the Web Services Input to the left input the **Join Data** module so that it will use the same schema as the **Enter Data Manually** module.
8. Configure the **Enter Data Manually** module as follows to provide a header and some test data to define the input schema of the web service (you can copy and paste the data from **Manual Input.txt** in the **Lab03** folder):
    - **DataFormat**: CSV
    - **HasHeader**: Checked
    - **Data**:
      PatientID,Pregnancies,PlasmaGlucose,DiastolicBloodPressure,TricepsThickness,SerumInsulin,BMI,DiabetesPedigree,Age
      1882185,9,104,51,7,24,27.36983156,1.350472047,43
      1662484,6,73,61,35,24,18.74367404,1.074147566,75
      1228510,4,115,50,29,243,34.69215364,0.741159926,59
9. Verify that the top part of your experiment now looks like this:



10. Save and run the experiment to verify that there are no errors as a result of changing the input.
11. Visualize the output of the **Score Model** module, which determines the output of the web service, and verify that is contains only three rows – one for each test row in the **Enter Data Manually** module.

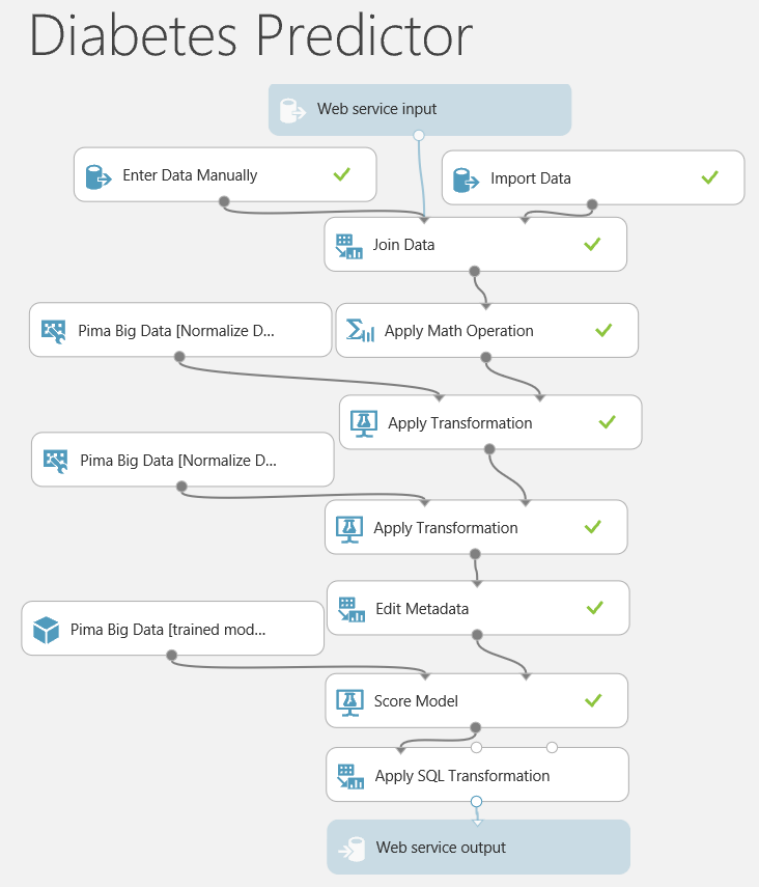
Note: The output includes all columns in the data flow. You will now modify the experiment so that only the patient ID, physician, diabetic prediction, and probability are returned.

12. Delete the connection between the **Score Model** module and the **Web Services Output**.
13. Add an **Apply SQL Transformation** module to the experiment and connect the output of the **Score Model** module to its **Table1** (leftmost) input.

14. Configure the **Apply SQL Transformation** module properties to update the SQL query script to the following code (which you can copy and paste from **Diabetes SQL.txt** in the **Lab03** folder):

```
SELECT PatientID,
       Physician,
       [Scored Labels] AS DiabetesPrediction,
       [Scored Probabilities] AS Probability
FROM t1;
```

15. Connect the output of the **Apply SQL Transformation** module to the input of the **Web service output**.
16. Verify that, with a bit of rearranging, your experiment resembles the figure below:



17. Save the run the experiment.
18. Visualize the output from the **Apply SQL Transformation** module and verify that only four columns are returned.

Add a Parameter

Note: The predictive experiment imports reference data from Azure SQL Database using the connection properties that are set for the **Import Data** module. To increase flexibility, you can configure these as parameters so that client applications can specify alternative connection settings.

1. Select the **Import Data** module that imports the physician data from Azure SQ Database, and in the **Properties** pane, above the **Database server name**, click the ☰ icon and select **Set as web service parameter**.

2. Scroll to the bottom of the **Properties** pane and under **Web Service Parameters**, note the **Database server name** parameter is listed. Then click its ☰ icon and select **Provide default value**.
3. In the text box for the default value, type the name of your Azure SQL Database server.
4. Save and run the experiment.

Deploy the Web Service

You are now ready to publish and test the web service.

1. With the **Diabetes Predictor** predictive experiment open, in the **Deploy Web Services** menu, click **Deploy Web Service [New] Preview**. A web page with a header **Deploy "Diabetes Predictor" experiment as a web service** will appear.
2. Set the web service configuration as follows:
    - **Web Service Name**: DiabetesPredictor
    - **Storage Account**: The Azure storage account associated with your workspace
    - **Pricing Plan**: The Azure Machine Learning pricing plan you created with your workspace.
3. Click the **Deploy** box in the lower left of the page. After some time, the **Quickstart** page for your **Diabetes Predictor** web service should appear.

Consume the Web Service

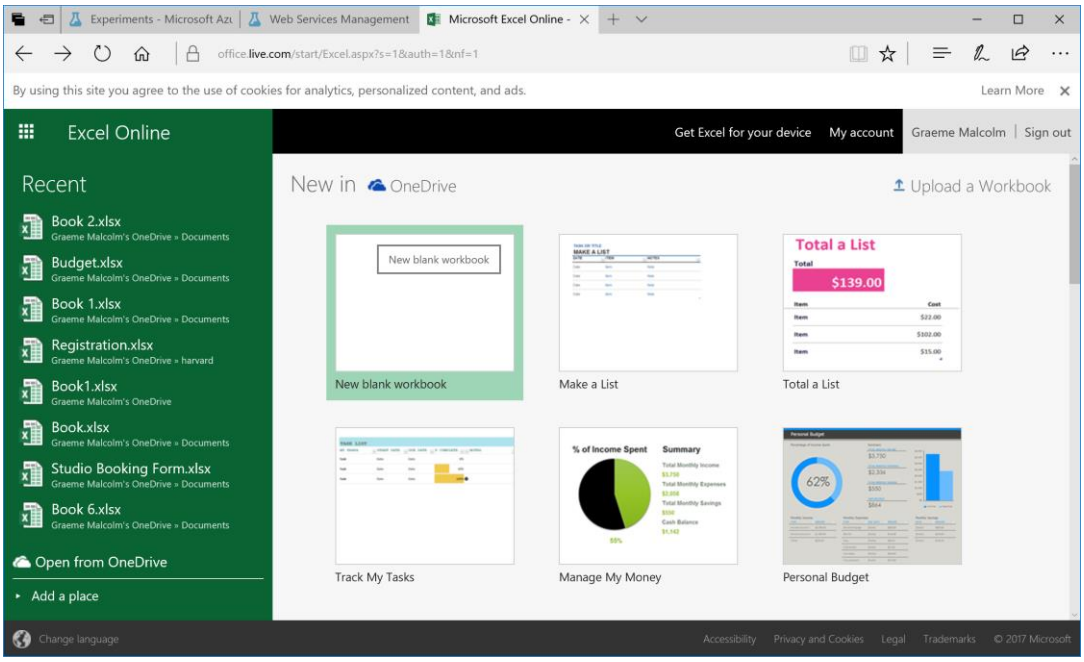
Now that you have published the predictive web service, you can consume it from client applications.

Note: In this exercise, you will consume the web service from Excel Online, which is a service included with your Microsoft account (for example, an *outlook.com* account). If you have Excel 2010 or Excel 2013 installed on your local computer, you can use that instead of Excel Online if you prefer.

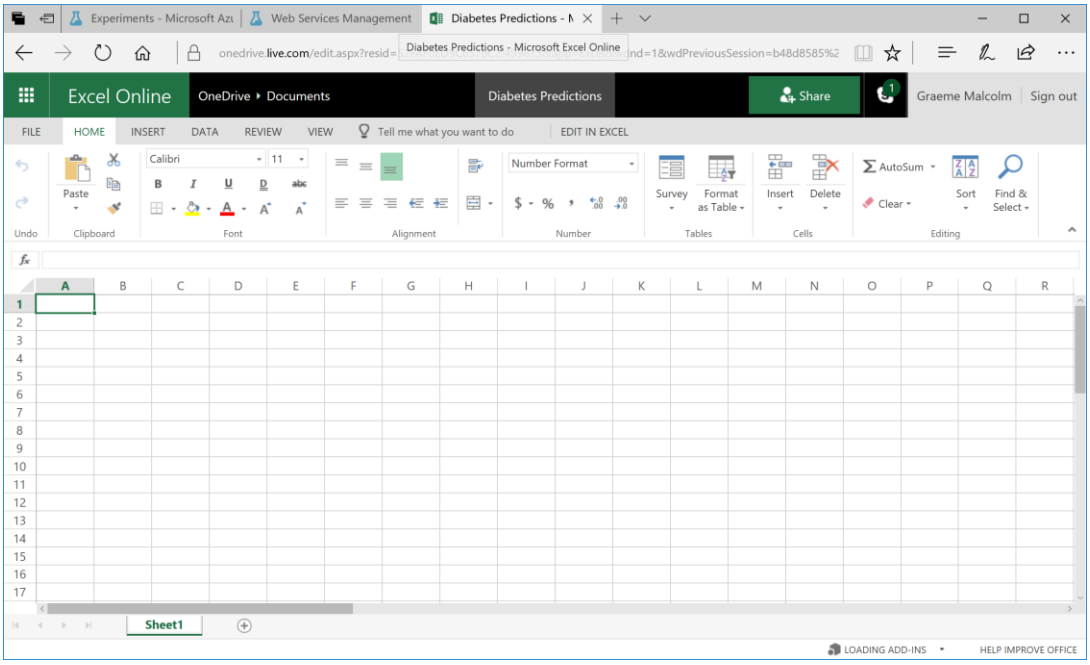
1. Click the **Configure** page for the web service, and next to **Sample Data Enabled?**, click **Yes**. Then click **Save** and wait for the changes to be saved.

    Note: By enabling sample data, client applications can download the sample data you defined in the **Enter Data Manually** module in the predictive experiment in order to test the web service.

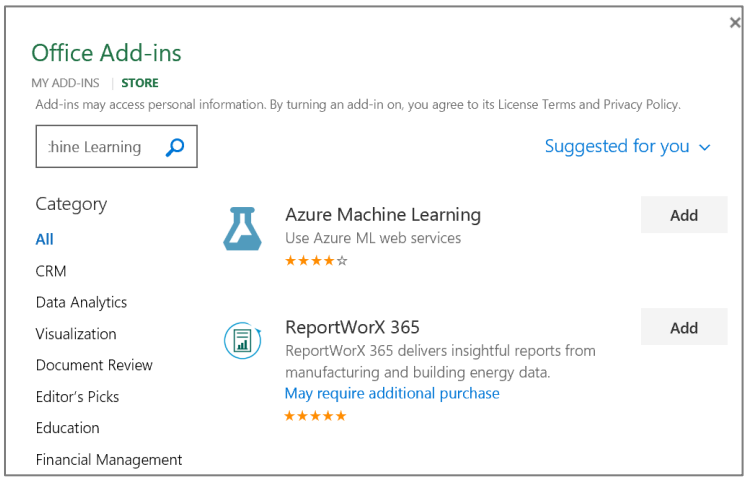
2. In the **Configure** page for your web service, click **Consume**, and note that this page lists the primary and secondary keys for your web service, and its request-response and batch endpoint URLs.
3. Open a new tab in your browser, navigate to https://office.live.com/start/Excel.aspx, and sign in using your Microsoft account.
4. Create a new blank workbook, as shown here:



5. At the top of the blank workbook, change the name to **Diabetes Predictions** as shown here:

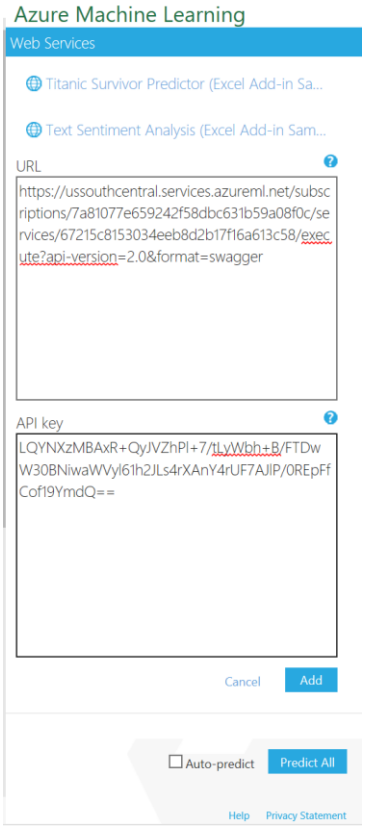


6. On the **Insert** tab, click **Office Add-ins**. Then in the **Office Add-ins** dialog box, select **Store**, search for *Azure Machine Learning*, and add the **Azure Machine Learning** add-in as shown below:

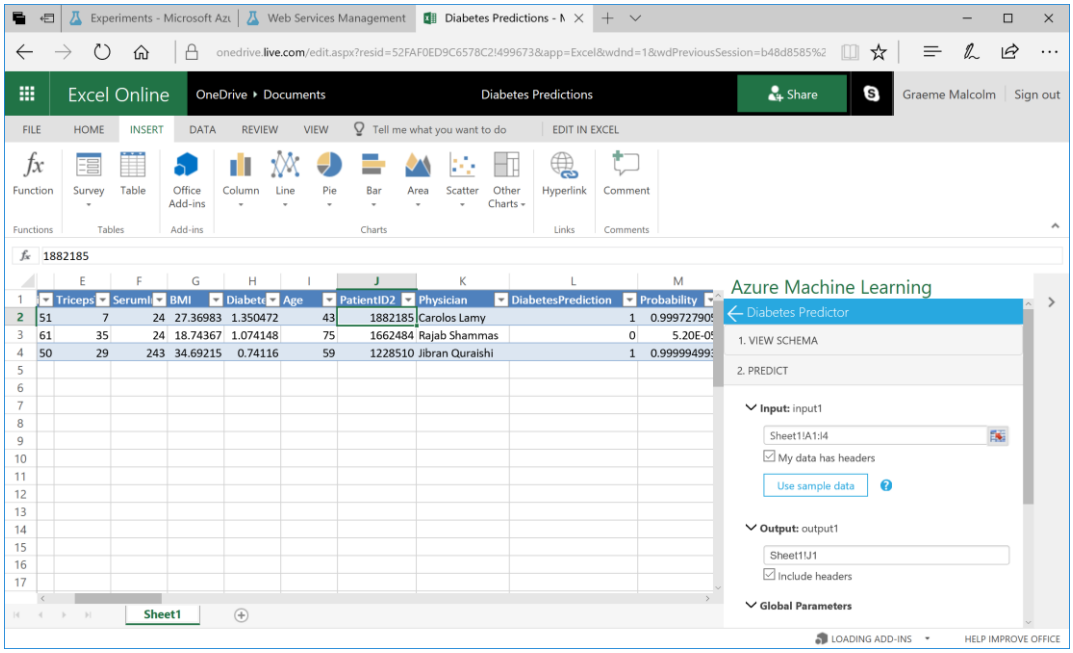


After the add-in is installed, it will appear as a pane named **Azure Machine Learning** on the right side of your Excel workbook.

7. In the **Azure Machine Learning** pane, click **Add Web Service**. Boxes for the URL and API key of the web service will appear.
8. On the browser tab containing the **Consume** page for your web service, click the **Copy** button for the **Request-Response** URL. Then return to the browser tab containing the Excel Online workbook and paste the URL into the URL box.
9. On the browser tab containing the **Consume** page for your web service, click the **Copy** button for the **Primary Key**. Then return to the browser tab containing the Excel Online workbook and paste it into the **API key** box.
10. Verify that the **Azure Machine Learning** pane in your workbook now resembles this, and click **Add**:



11. After the web service has been added, in the **Azure Machine Learning** pane, click **1. View Schema** and note the *inputs* expected by the web service, the *outputs* returned by the web service, and the *global parameters* for the web service.
12. In the Excel worksheet select cell A1. Then in the **Azure Machine Learning** pane, collapse the **1. View Schema** section and in the **2. Predict** section, click **Use sample data**. this enters sample input values in the worksheet.
13. With the cells containing the sample input data (cells A1 to I4) selected, in the **Azure Machine Learning** pane, click the button to select the input range and confirm that it is **'Sheet1'!A1:I4**.
14. Ensure that the **My data has headers** box is checked.
15. In the **Output** box type **J1**, and ensure the **Include headers** box is checked.
16. In the **Database server name** box, enter the name of your Azure SQL Database server.
17. Click the **Predict** button, and after a few seconds, view the predictions in cells J2:M4, as shown here:



18. Try changing a few of the input variables and predicting the diabetes classifications. You can add multiple rows to the input range and try various combinations.

Summary

In this lab, you have published predictive web services based on Azure Machine Learning experiments, and you have tested these web services using the test interface in the Azure Machine Learning Web Services Management portal, and from a client application (Excel).

Now that you know how to operationalize predictive web services, you are ready to incorporate them into big data processing operations.