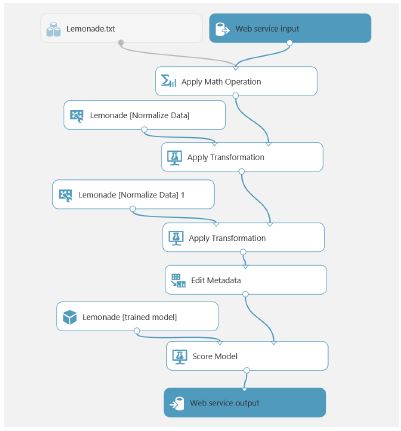
Exercise 2: Publishing and Using a Machine Learning Model

Now that you have trained a machine learning model, you can publish it as a web service and use it to predict labels from new feature data. In Azure Machine Learning Studio, you do this by creating a *predictive experiment* that encapsulates your model and the data preparation steps you have defined, and which defines the input and output interfaces through which features are passed into the model and predicted labels are returned. You then publish this predictive experiment as a web service in Azure.

Create a Predictive Experiment

1. In the **Lemonade Training** experiment, on the toolbar under the experiment canvas, click the **Set Up Web Service** icon and click **Predictive Web Service [Recommended]**. Then wait for the predictive experiment to be created and click **Close** to close the notification.

2. In the **Predictive Experiment** tab, change the experiment name from **Lemonade Sales Training [Predictive Exp.]** to **Predict Lemonade Sales**. Then rearrange the modules in the predictive experiment



The experiment consists of:

• A web service input with a schema defined by the original **Lemonade-Sales.csv** training dataset.

* 1. • The **Apply Math** operation to replace **Rainfall** with its natural log.
  2. • An **Apply Transformation** module that normalizes the **Temperature** and **Flyers** features using the ZScore statistics from the training data.
  3. • An **Apply Transformation** module that normalizes the **Rainfall** and **Price** features using the MinMax statistics from the training data.
  4. • An **Edit Metadata** module that clears the **Day**, **Date**, and **Sales** features.
  5. • A **Score Model** module that predicts the scored label from the input data by applying the rained model.
  6. • A web service output that returns the results to the calling application.

3. Delete the **Lemonade-Sales.csv** dataset, then search for an **Enter Data Manually** module, add it to the top of the experiment, and connect its output to the input of the **Apply Math Operation**

The **Lemonade-Sales.csv** dataset included the **Sales** field, which is what the model predicts. It therefore makes sense to redefine the input schema for the web service so that the **Sales** field is not submitted.

4. Select the **Enter Data Manually** module, and in the **Properties** pane, ensure **DataFormat** is set to **CSV** and **HasHeader** is selected, and then enter the following test data (which you can copy and paste from **Input.txt** in the folder where you extracted the lab files):

Date,Day,Temperature,Rainfall,Flyers,Price

01/01/2017,Sunday,27,2.00,15,0.3

02/01/2017,Monday,28.9,1.33,15,0.3

03/01/2017,Tuesday,34.5,1.33,27,0.3

04/01/2017,Wednesday,44.1,1.05,28,0.3

5. Select the **Edit Metadata** module and edit its properties to launch the column selector and remove the **Sales** field – this field no longer exists in the input dataset, so referencing it here will cause a runtime error when the web service is called.

6. Run the experiment.

7. Visualize the output from the **Score Model** module, and note that it includes all of the fields from the input data you entered manually along with the scored labels.

Client applications calling the web service only require the scored labels, so you can modify the output schema to remove the other fields.

8. Select the connection between the **Score Model** and **Web service output**, and delete it.

9. Search for and add an **Apply SQL Transformation** module, connecting it to the dataflow between the Score Model and Web service output modules

10. Modify the properties of the **Apply SQL Transformation** module to replace the default SQL Query Script with the following code:

SELECT [Scored Labels] AS PredictedSales

FROM t1;

11. Run the experiment.

12. Visualize the output of the **Apply SQL Transformation** module and verify that only the **PredictedSales** column is now returned.

Deploy the Web Service

1. At the bottom of the experiment canvas, click **Deploy Web Service**, and wait for the web service to be deployed.

2. In the **predict lemonade sales** web service page, click the **New Web Services Experience** link, which opens a new tab with the Azure Machine Learning Web Services Management portal

3. Click the **Consume** tab, and note that this page contains the keys and endpoint URLs that are used by a client application to call the web service.

Consume the Web Service

1. In a new browser tab, navigate to https://office.live.com/start/Excel.aspx and sign in using your Microsoft credentials.

2. Create a new blank workbook.

3. On the Insert tab of the ribbon, click **Office Add-ins**, and on the **Store** tab, search for **Azure Machine Learning**

4. Add the **Azure Machine Learning** add-in. This opens the **Azure Machine Learning** tab in Excel

The add-in includes links for some built-in sample web services, but you will add your own web service.

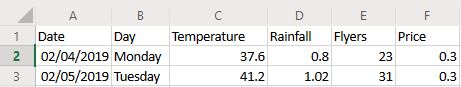
5. Click **Add Web Service**.

6. Switch back to the Web Services Management tab in your browser, and copy the **Request-Response** URL to the clipboard. Then return to the Excel Online tab and paste the copied URL into the **URL** textbox of the Azure Machine Learning pane

7. Switch back to the Web Services Management tab in your browser, and copy the **Primary Key** to the clipboard. Then return to the Excel Online tab and paste the copied key into the **API key** textbox of the Azure Machine Learning pane

8. Click **Add** to add the web service.

9. Starting in cell A1, enter the following data in the worksheet:



10. In the Azure Machine Learning pane, in the **Input** box, enter **A1:F3** and ensure that the **My data has headers** checkbox is checked.

11. In the **Output** box, enter **G1** and ensure that the **Include headers** checkbox is selected.

12. Click **Predict**

13. Wait for the web service to be called, and then view the **PredictedSales** values that are returned

*Challenge*

Try predicting sales for today and tomorrow if Rosie increases the number of flyers to 100.