Microsoft Data Science

Cortana Intelligence Suite – Foundations for Dynamics

Student Lab

Last Updated: Ryan Swanstrom, 9/8/2016

In this lab, you will gain experience using Azure Storage, Azure ML, Azure Data Factory, and Power BI to create recommendations using data from Microsoft Dynamics.

Part 1 – Environment Configuration

Lab Steps

A. Account Activation – skip it

1. Activate your Azure Subscription
    i. Open the Portal
    ii. Create one empty Resource Group
2. Explain a situation where data was used in a new and unexpected way in a business or organization
3. List three advantages to using a cloud or hybrid architecture
4. List three objections to hosting an application in the cloud, and three responses to those objections

B. Access the VM

1. Log in to the virtual machine with details provided by the instructors

C. Create Storage Account

1. Create a Storage Account in the region closest to the class location – note the name and access keys
    i. Account Kind: **General purpose**
    ii. Resource Group: **Create New: <first-last-name>dxtraining**
2. Connect to http://studio.azureml.net and create a free account for the class

Part 2 – Data Discovery and Ingestion

Lab Steps

A. Azure Data Catalog – skip it

B. Azure Storage

1. Open the Azure Portal, locate your storage account (or create one if you haven't, call it **<first-last-name>dxfiles**), and a container (or create one if you have not, call it **rawdata**). Note the name of your storage account, container, and storage key
2. Use Microsoft Azure Storage Explorer to upload the files specified by the instructor to the container within your storage account
    a. ProductCatalog.csv
    b. UsageFile1.txt
    c. UsageFile2.txt
    d. UsageFile3.txt
    e. UsageFile4.txt
    f. UsageFile5.txt

C. Exploring Your Data

1. Use R, Excel, Azure ML or any other exploration tools you've seen in the class to explore the shape, size, layout, distribution and other characteristics you can find in the data
2. Document that in any format and be prepared to discuss

Microsoft Data Science

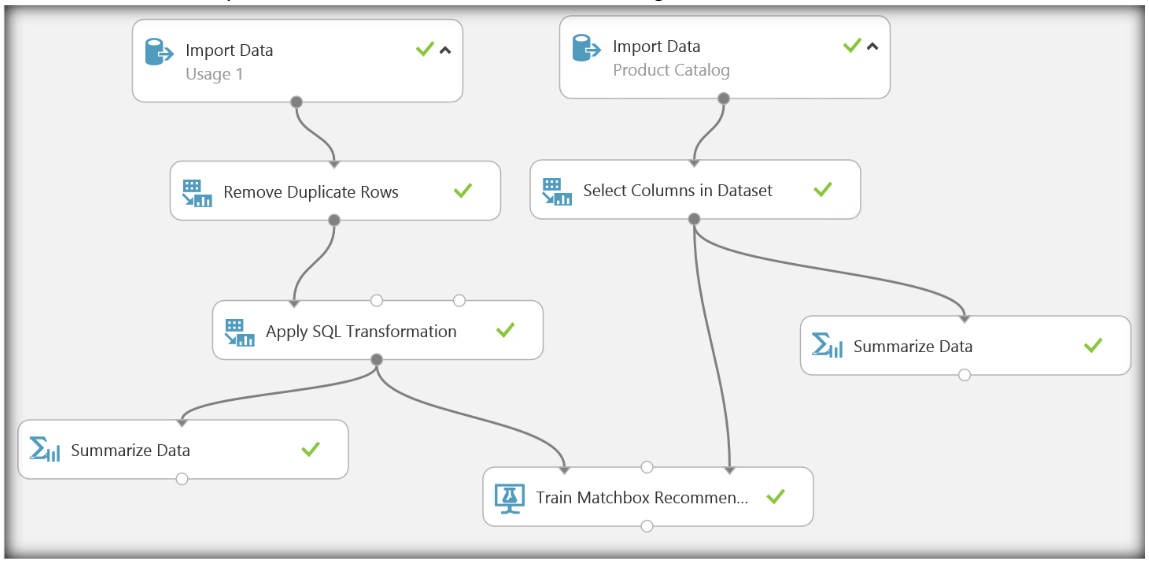
Part 3 – Modeling for Machine Learning

Lab Steps

A. Create and Run an Experiment in Azure ML

1. Visit http://studio.azureml.net and log in
2. Click the large **+ NEW** in the bottom and select **Blank Experiment**
3. Click on the title and change it from *Blank Experiment created on* … to **DX Recommendations**
4. Import the Data
    i. Drag an **Import Data** module into your workspace (hint: use the search bar if you cannot find it)
    ii. On the right-hand side, set the properties to import the **UsageFile1.txt** you just added to Azure Storage
        1. Data source: **Azure Blob Storage**
        2. Authentication type: **Storage Account**
        3. Account name: **<first-last-name>dxfiles (if that is what you named your storage account)**
        4. Account key: <get this from the Portal under Access keys or right-click in Microsoft Azure Storage Explorer>
        5. Path: **/rawdata/UsageFile1.txt**
        6. Blob file format: **CSV**
        7. File has header row: **do NOT check**
        8. Use cached results: **Check this**
    iii. Right-click the module -> **Edit Comment,** type **Usage File 1**
5. Repeat the import steps above for the **/rawdata/ProductCatalog.csv** with the comment **Product Catalog**
6. Drag in a **Remove duplicate rows** module and connect to the output of the Usage File 1 module.
    i. Column names: **Col1, Col2**
    ii. check **Retain first duplicate row**
7. Drag in a **Select columns in Dataset** module, and connect it with the output of the Product Catalog module
    i. On the right-hand side, click **Launch column selector**
    ii. For the SELECTED COLUMNS: **Col1, Col2, Col3**

8. Drag in an **Apply SQL Transformation** module and connect the left most input (Table 1) with the output from the remove duplicate rows module [this step adds artificial ratings to the dataset]
   i. Type in the following for the SQL Query script
      select col1, col2, abs(random() % 5)
      from t1;
9. Drag in two **Summarize Data** modules and connect one with each of the previous two modules created
10. Click **Run** at the bottom of the page
11. Wait for green checkmarks, then right-click on either **Summarize Data** module, **Results dataset -> Visualize**
12. Drag in a **Train Matchbox Recommender** and connect the Apply SQL Transformation module to the left-most input and the Select columns in dataset module to the right-most input
13. Click **Run**. The experiment should look like the diagram below.

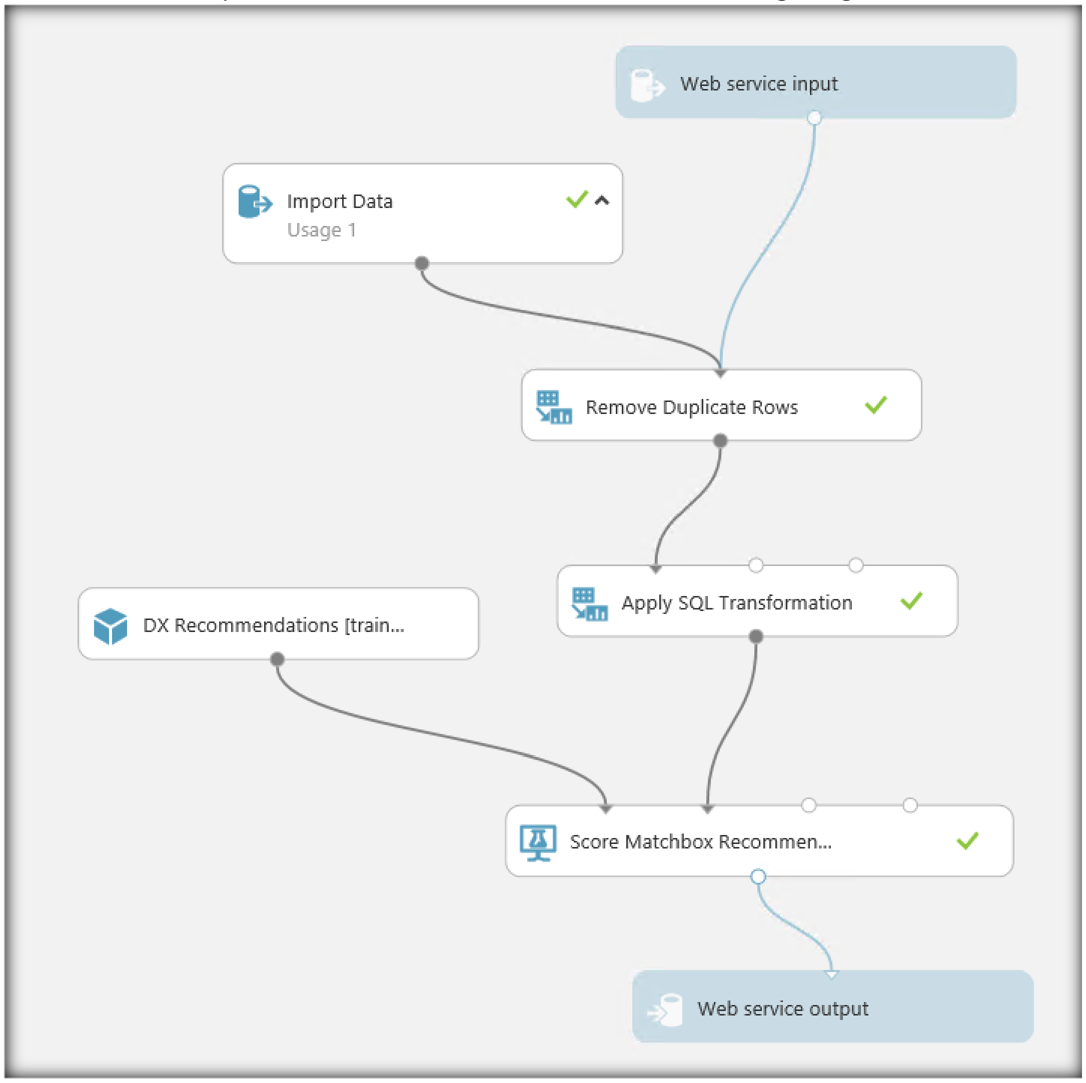


B. Turn an Azure ML Experiment into a web service

1. At the bottom of the screen, **SET UP WEB SERVICE -> Predictive Web Service**
2. Replace the **Score** module with a **Score Matchbox Recommender** module
   i. The “trained model” should connect to the first input
   ii. The **Apply SQL Transformation** module should connect to the second input
   iii. Connect the output to the web service output module
   iv. Change the following settings for the module
      1. Recommender prediction kind: **Item Recommendation**
      2. Recommended item selection: **From All Items**

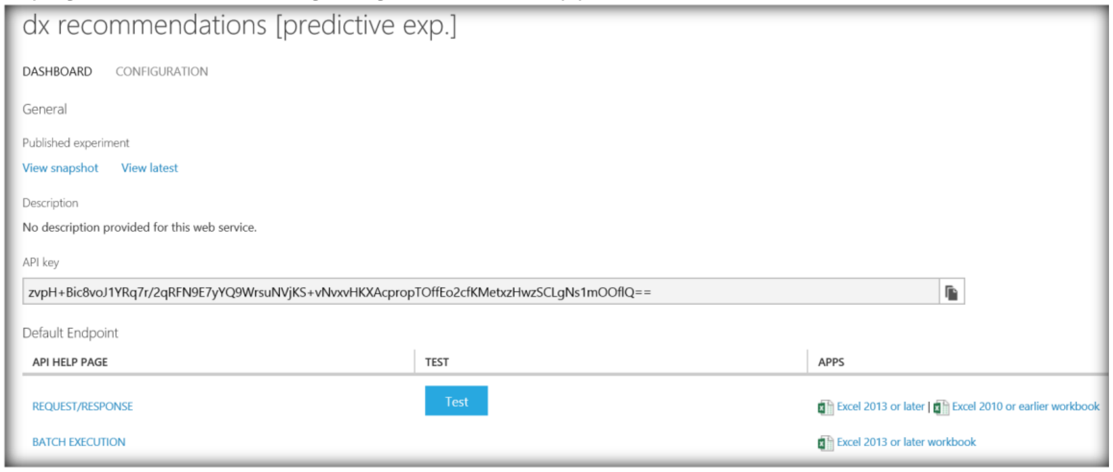
    3. Maximum number of items ... : **5**

3. Click **Run**
4. The Predictive Experiment should now look like the following diagram



5. Click **DEPLOY WEB SERVICE** at the bottom

6. A page like the following diagram should appear



dx recommendations [predictive exp.]

DASHBOARD CONFIGURATION

General

Published experiment

View snapshot View latest

Description

No description provided for this web service.

API key

zvpH+Bic8voJ1YRq7r/2qRFN9E7yYQ9WrsuNVjKS+vNvxvHKXAcpropTOffEo2cfKMetxzHwzSCLgNs1mOOflQ==

Default Endpoint

API HELP PAGE	TEST	APPS
REQUEST/RESPONSE	Test	Excel 2013 or later | Excel 2010 or earlier workbook
BATCH EXECUTION		Excel 2013 or later workbook

7. Click the **Test** button
8. Try placing the following user account numbers in **COL1** and clicking the check mark (note, one number at a time)
   i. 1158758915044079616
   ii. 4324439393806021632
   iii. 4205789353816416256
9. Five item recommendations should appear for each user account number

Microsoft Data Science

Part 4 – Evaluating the Model

Lab Steps

A. No Separate Lab

Part 5 – Data Preparation (ADF)

Lab Steps

Note: JSON prettifier may be useful: http://jsoneditoronline.org

Prework – Create an input file to recommend

1. On the VM, create a copy of the file UsageFile1.txt and rename it **inputFile.csv**
2. Add a line to the top of the file (use Notepad) **Col1,Col2,Col3,Col4,Col5**
3. Save the file
4. Upload the file to the **rawdata** folder in Azure Storage

A. Create the ADF

1. Do the Prework above
2. Open Portal: **https://ms.portal.azure.com/#**
3. After logging into the Azure Portal, do the following:
    a. Click **NEW** on the left menu.
    b. Click **Data + Analytics** in the **New** blade.
    a. Click **Data Factory** on the **Data + Analytics** blade.
2. In the New data factory blade, enter **<yourfirstandlastname>dxDF** for the Name.
3. Select the Azure subscription where you want the data factory to be created.
4. Select existing resource group from Part 1, named: **<first-last-name>dxtraining**
5. Click **Create** on the New data factory blade.
6. You will see the data factory being created in the **Startboard** of the Azure Portal

B. Create the Linked Service

1. Click **Author and deploy** on the **DATA FACTORY** blade for **dxDF**. This launches the Data Factory Editor.
2. Click **New data store** and choose **Azure storage** - You should see the JSON script for creating an Azure Storage linked service in the editor.

3. Replace <accountname> with the name of your Azure storage account and <accountkey> with the access key of the Azure storage account.
4. Click **Deploy** on the command bar to deploy the storage linked service - After the linked service is deployed successfully, the Draft-1 window should disappear and you will see **AzureStorageLinkedService** in the tree view on the left.
5. In the **Data Factory Editor**, click **New compute** on the command bar and select **Azure ML**
6. Specify <mlEndpoint> with the Request URI from the BATCH EXECUTION page of the predictive experiment from Part 3, don't include the **?api-version=2.0** at the end of the URL
7. Specify <apiKey> with the API key of the predictive experiment.
8. Delete the updateResourceEndpoint line and the preceding comma
9. Click **Deploy** on the command bar to deploy the linked service.
10. Confirm that you see both **AzureStorageLinkedService** and **AzureMLLinkedService** in the tree view on the left.

C. Create Datasets

(Input Dataset)

1. In the **Data Factory Editor**, click **New dataset** on the command bar and select **Azure Blob storage**.
2. Copy and paste the snippet below to the **Draft-1** window. In the JSON snippet, you are creating a dataset called **BlobInput** that represents input data for an activity in the pipeline. In addition, you specify that the input data is in the blob container called **rawdata**.

```
-------------------

{

   "name": "InputBlob",

   "properties": {

      "published": false,

      "type": "AzureBlob",

      "linkedServiceName": "AzureStorageLinkedService",

      "typeProperties": {
          "fileName": "inputFile.csv",

          "folderPath": "rawdata",

          "format": {

              "type": "TextFormat",

              "columnDelimiter": ","

          }

      },

      "availability": {

          "frequency": "Day",

          "interval": 1

      },

      "external": true,

      "policy": {

          "externalData": {

              "retryInterval": "00:01:00",

              "retryTimeout": "00:10:00",

              "maximumRetry": 3

          }

      }

   }

}
```

----

3. Click **Deploy** on the command bar to deploy the newly created input dataset. You should see the dataset in the tree view on the left.

(Output dataset)

Now, you will create the output dataset to represent the output data stored in the Azure Blob storage.

4. In the **Data Factory Editor**, click **New dataset** on the command bar and select **Azure Blob storage**.
5. Copy and paste the snippet below to the **Draft-1** window. In the JSON snippet, you are creating a dataset called **BlobOutput**, and specifying the structure of the data that will be produced by the Hive script. In addition, you specify that the results are stored in the blob container called **rawdata** and the folder called **recommendations**. The availability section specifies that the output dataset is produced monthly.

```
{
   "name": "ResultBlob",
   "properties": {
      "published": false,
      "type": "AzureBlob",
      "linkedServiceName": "AzureStorageLinkedService",
      "typeProperties": {
         "fileName": "{filepart}result.csv",
         "folderPath": "rawdata/recommendations/{folderpart}/",
         "format": {
            "type": "TextFormat",
            "columnDelimiter": ","
         },
         "partitionedBy": [
            {
               "name": "folderpart",
               "value": {
                  "type": "DateTime",
                  "date": "SliceStart",
                  "format": "yyyyMMdd"
               }
            },
            {
               "name": "filepart",
               "value": {
                    "type": "DateTime",
                    "date": "SliceStart",
                    "format": "HHmmss"
                  }
                }
              ]
            },
            "availability": {
              "frequency": "Day",
              "interval": 15
            }
          }
        }
```

----

6. Click **Deploy** on the command bar to deploy the newly created dataset.
7. Verify that the dataset is created successfully.

D. Create Pipelines

1. In the **Data Factory** Editor, click ellipsis (…) **More commands** and the click **New pipeline**
2. Copy and paste the snippet below to the **Draft-1** window.

```
        {
          "name": "RecommendationPipeline",
          "properties": {
            "description": "use AzureML model to recommend items",
            "activities": [
              {
                "type": "AzureMLBatchExecution",
                "typeProperties": {
                  "webServiceInput": "InputBlob",
                  "webServiceOutputs": {
                    "output1": "ResultBlob"
                },
                "webServiceInputs": {},
                "globalParameters": {}
            },
            "inputs": [
                {
                    "name": "InputBlob"
                }
            ],
            "outputs": [
                {
                    "name": "ResultBlob"
                }
            ],
            "policy": {
                "timeout": "02:00:00",
                "concurrency": 3,
                "executionPriorityOrder": "NewestFirst",
                "retry": 1
            },
            "scheduler": {
                "frequency": "Day",
                "interval": 15
            },
            "name": "MLActivity",
            "description": "prediction analysis on batch input",
            "linkedServiceName": "AzureMLLinkedService"
        }
    ],
    "start": "2015-02-13T00:00:00Z",
    "end": "2015-02-14T00:00:00Z",
    "isPaused": false,
    "pipelineMode": "Scheduled"
  }
}
```

----

3. Click **Deploy**

E. Monitor Pipelines

1. Click **X** to close Data Factory Editor blades and to navigate back to the **Data Factory** blade, and click on **Diagram**.
2. In the **Diagram View**, you will see an overview of the pipelines, and datasets used in this tutorial.
3. To view all activities in the pipeline, right-click on pipeline in the diagram and click **Open Pipeline**.
4. In the **Diagram View**, double-click on the dataset **BlobInput**. Confirm that the slice is in **Ready** state. It may take a couple of minutes for the slice to show up in Ready state. If it does not happen after you wait for some time, see if you have the input file (**inputFile.csv**) placed in the right container (**rawdata**).
5. Click **X** to close **BlobInput** blade.
6. In the **Diagram View**, double-click on the dataset **ResultBlob**. You will see that the slice that is currently being processed.
7. When processing is done, you will see the slice in **Ready** state.
8. In Azure Storage, the rawdata container should now have a **recommendations** folder with a .csv file in it

Microsoft Data Science

Part 6 – Deployment

Lab Steps

A. Analyzing data in Power BI

1. ?