Workshop: ICP for Data – Analyze

Lab 1 – Define Project and Set Up Data Connections

*May 29, 2018*

*Author: Rui Fan* [*rui.fan@ibm.com*](mailto:rui.fan@ibm.com)

*Reference assets from: Sidney Phoon* [*yfphoon@us.ibm.com*](mailto:yfphoon@us.ibm.com)

**Table of contents**

Contents

Overview

**Required software, access and files**

1. Log into DSXL on ICP for Data (ICP4D)

2. Set up the lab project

3. Define data source connections

4. Verification steps: Read data from remote data sets and local files

# **Overview**

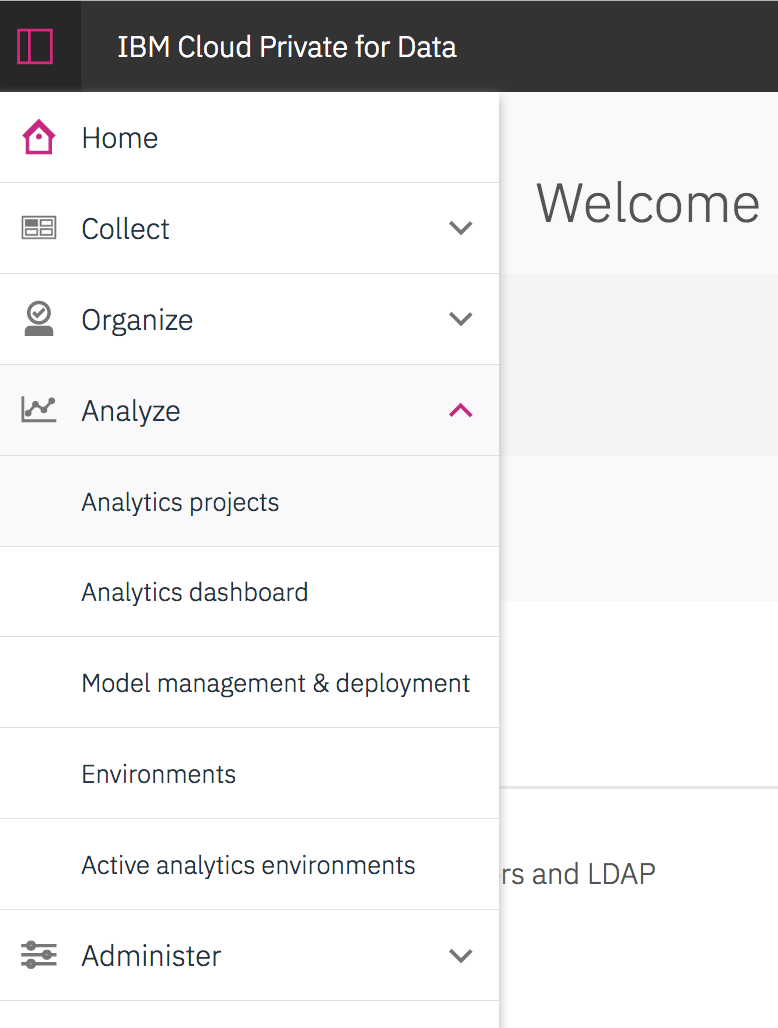
In this lab you will learn how to get access to DSX Local in ICP for Data(ICP4D) to build model and deploy.

# **Required software, access, and files**

* To complete this lab, you will need access to a IBM Cloud Private for Data environment.
* You will also need to download and unzip this GitHub repository: <https://github.com/Rui425/ICP4D-/blob/master/DSXL%20Projects/ICP4D%20Telco%20Churn.zip>

# **1: Log into DSXL on ICP for Data (ICP4D)**

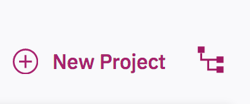
1. Log in ICP4D environment using the given credentials
2. **DSXL** refers to the context of ‘**Analyze’**. Click ‘**Analytics projects**’



You will see a project view of different project you worked on.

# **2: Set up the lab project**

1. Click ‘New Project’ to begin your journey in DSXL for ICP4D.



1. To create this project select **‘from file’**. Under ‘Notebook File’ drag and drop or browse for the Zip file you downloaded earlier (ICP4D Telco Churn.zip).
2. Name the Project to whatever you want.
3. Click ‘**Create**’, then the project is created.
4. Click ‘**Assets**’ to view all available assets that are in this project.

# **3: Define data source connections**

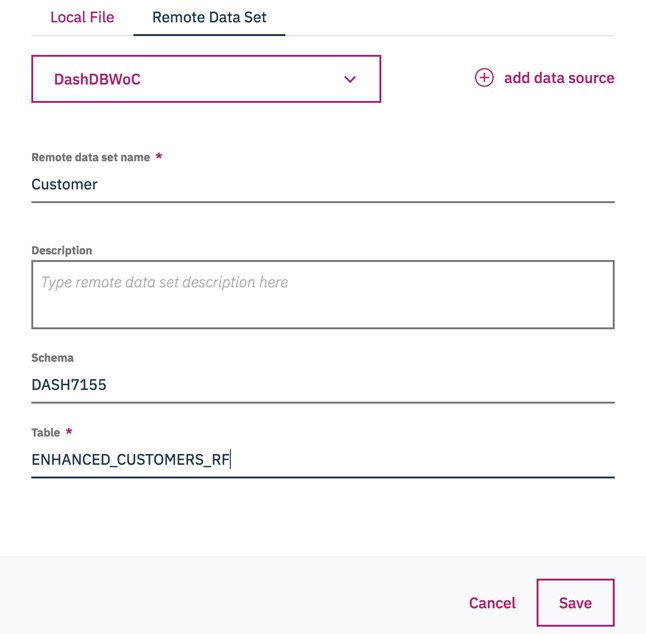
In this workshop, we want to make use of data from different sources. We will be initially connecting to the merged data set of customer demographics and customer activity that you created earlier.

The first part is to create a data source connection to dashDB:

1. From the project folder navigate to the ‘**data sets**’ section and select ‘**add data set**’. Select the ‘**Remote Data Set**’ tab and click ‘**add data source**’ to define the data source connected.
2. Fill in the blanks with the corresponding data source information. Use the credentials for the environment that contains the merged data you created in the earlier labs. For the shared environment used for the training the credentials are:
   1. Name: Give it any name you want.
   2. Type : Select ‘dashDB’
   3. JDBC URL : jdbc:db2://dashdb-entry-yp-dal10-01.services.dal.bluemix.net:50000/BLUDB
   4. Username: dash7155
   5. Password: 4Km9h\_\_AgGgK
3. Click ‘**Create**’. The connection is created.

The second part is to define the specific data table we want to import into our project:

1. Navigate back to the ‘**Remote data set**’ tab and choose the data source you just added from the Pull down menue.
2. Fill in the blanks with the corresponding data table information:
   1. Name: Customer
   2. Schema: dash7155
   3. Table: Type in the table name that you created containing the merged customer demographics and activity **ENHANCED\_CUSTOMER\_{***YourInitials****}***

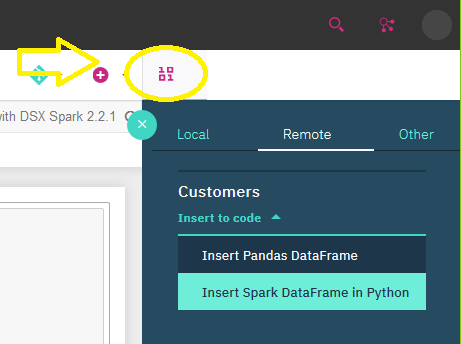


1. Click ‘**Save’**. The connected remote data is now listed in the ‘data set’ section on your projects home page

# **4: Verification steps: Create a new notebook to test reading data from remote data sets and local files**

This will also give you a brief understanding on what is a Notebook and how to load data into a notebook.

1. Under the ‘**Notebooks’** tab, create a new notebook by selecting ‘**add notebook**’. Ensure that the ‘**Blank’** tab is Selected. You will be creating a new blank notebook for this project.
   1. Name: enter in any name you would like.
   2. Environment: Select **‘Jupyter with Python3.5, Scala2.11, R3.4.3 Spark 2.2.2.1**’
   3. Language: Select **Python 3.5**.
   4. Click ‘**Create’**
2. We will now automatically generate the python code to test if we can import the data table from dashDB
   1. Click the blank in the first cell you see in the notebook.
   2. Click the '1010' button: ‘../../../../../Desktop/Screen%20Shot%202018-06-01%20at%2011. ‘ on the top right of the screen.
   3. Choose the **'Remote'** tab
   4. Select the ‘**Insert to code’** pull down menu under the ‘**Customer’** data connection we created earlier.
   5. Choose the **'Insert Spark DataFrame in Python'** option. Note the new Python code that is generated in the cell in your Notebook.

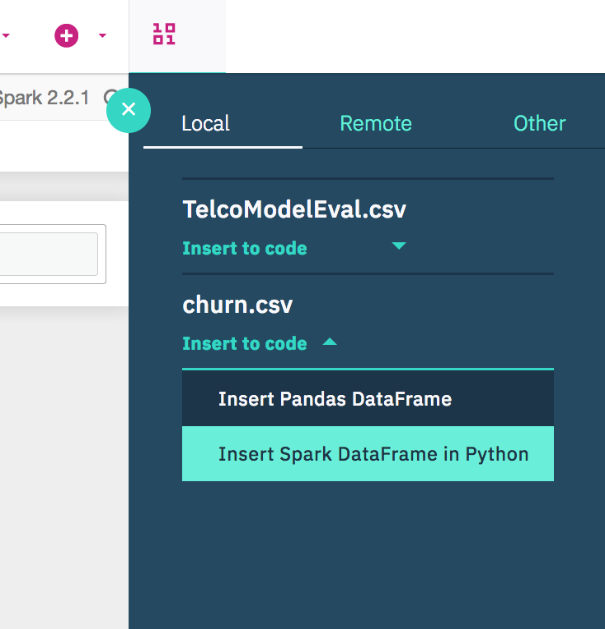


* 1. To Execute the code click the run button in the tool bar.



* 1. If a listing of the data field names and first 5 rows appears under the cell then you have correctly connected to a remote data source.

1. We will now automatically generate the Python code to test if we can load a local data set from a csv flat file.
   1. Create a new blank cell in your Jypyter Notebook
   2. Click the **'1010'** button: ‘../../../../../Desktop/Screen%20Shot%202018-06-01%20at%2011. ‘ on the top right of the screen.
   3. Choose the ‘**Local’** Tab
   4. Select the ‘**Insert to Code’** pull down manu under the file ‘**Churn.csv’**
   5. Select ‘**Insert Spark DataFrame in Python’** note the new python code that is automatically generated in the cell.



* 1. Click in the cell and click the run button in the toolbar to execute the code.



* 1. If a data table appears below the cell with the first 5 rows of the churn data then you have correctly connected to a local data source.