Workshop: ICP for Data - Analyze

# Lab 1

# Define Project and Set Up Data

# Connections

*September 25th, 2018*

*Author: Steven D. Reeves* [*sdreeves@us.ibm.com*](mailto:sdreeves@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

**5. Build and save Spark ML model with Juypter Notebook**

5.1 Review and execute the code in TelcoChurn\_SparkML notebook

5.2 Verify that the model is saved under the model section of the project

**6. Test, Batch Score and Evaluate saved model**

6.1 Test the model

6.2 Create a batch score

6.3 Create an evaluation

**7. Deployment: Commit and push project**

**8. Model Management & Deployment**

8.1 Create a Project release

8.2 In the Assets tab, create an online and batch deployment for the deployed Telco\_Churn\_ML\_model

8.3 Launch the project and enable

8.4 Click the online deployment created in step 8.2, and test the model in the API interface.

# **Overview**

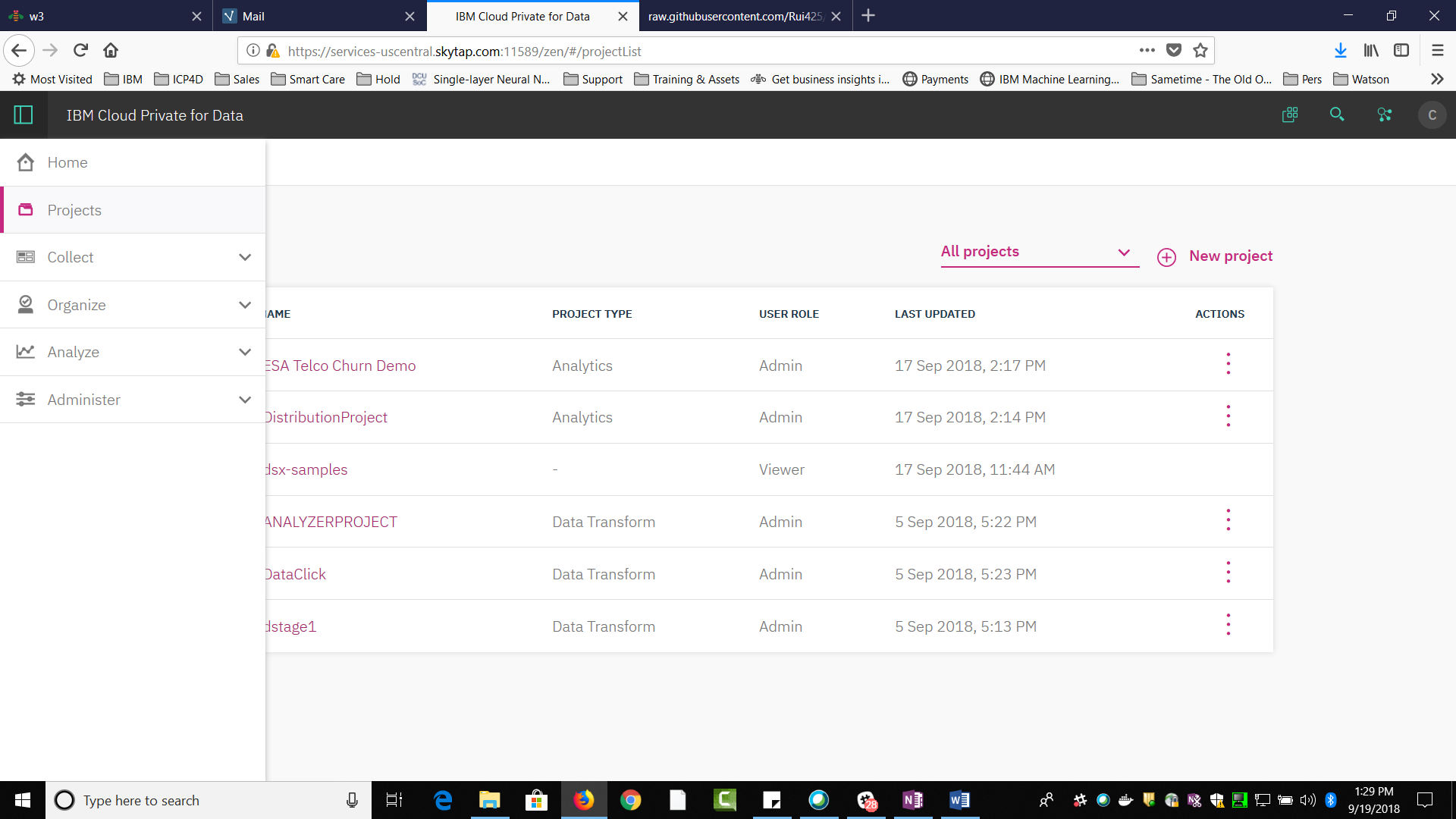
In this lab you will learn how to access DSX Local in ICP for Data(ICP4D) to build and deploy models.

# **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-/raw/master/Analyze%20Project/ICP4D%20Telco%20Churn-YourInitials.zip>

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

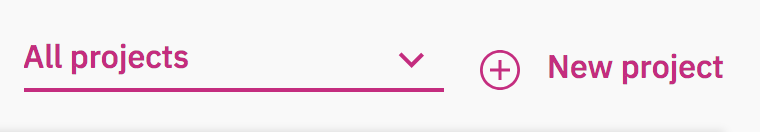
1. Log into the ICP4D environment using the class credentials provided.
2. The workflow for the Analyze process, starts with the creation of a Project. Click ‘**Projects’** to create your analytics project.



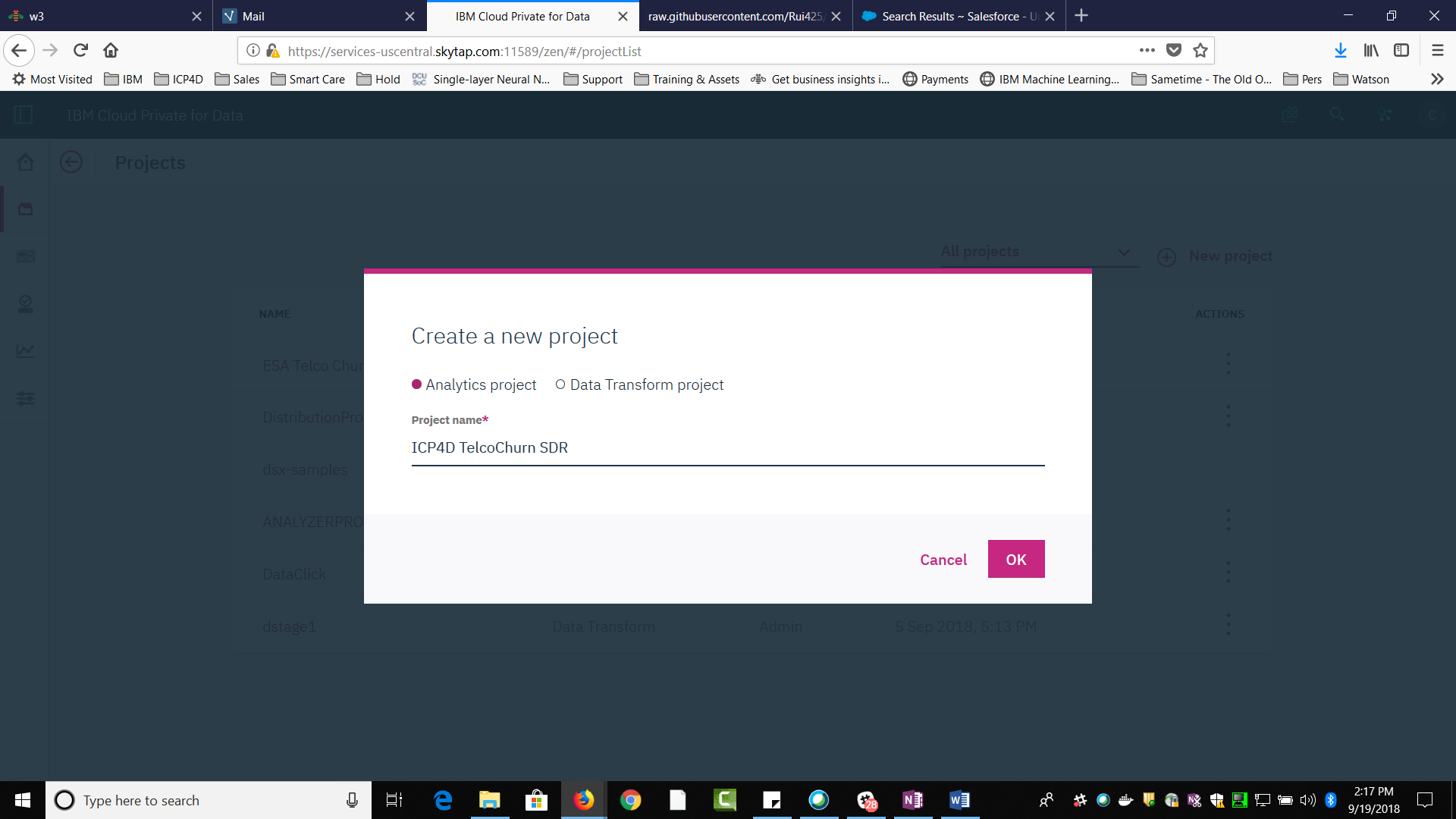
You will see a project view of different projects.

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

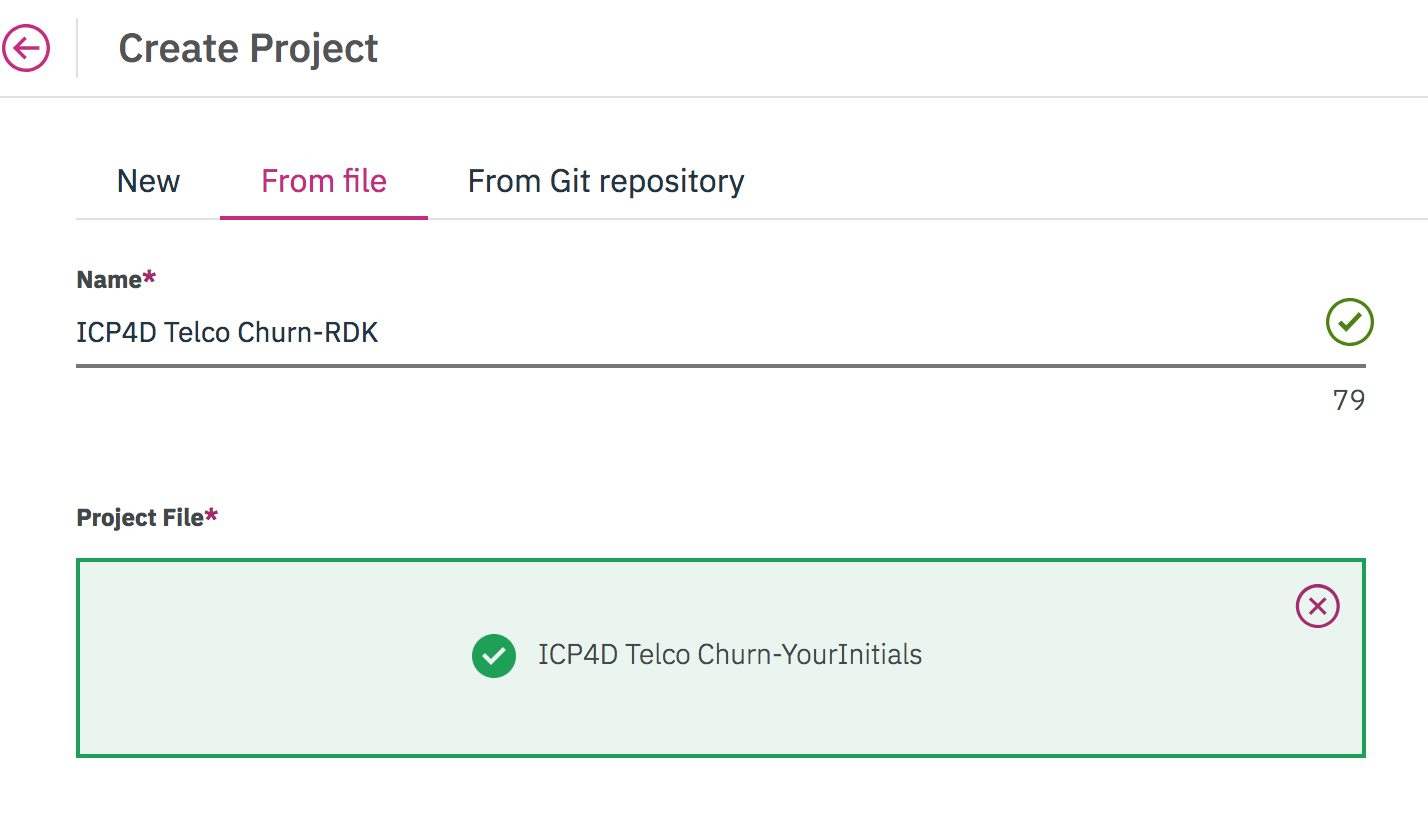
1. Click ‘New Project’ to begin the workflow in DSXL for ICP4D.



1. Select Analytics Project and name it something unique like ICP4D Telco Churn with your initials. Click OK.



1. Select **‘From file’**. Under ‘Project File’ drag and drop or browse for the Zip file you downloaded earlier (ICP4D Telco Churn-YourInitials.zip), and make sure the name remains unique.



1. Name the Project whatever you want. Make sure that it is unique for the class.
2. Click ‘**Create**’, then the project is created. **NOTE**: Assets will initially show a count of “0”. Click on “Projects” in the top left of the window and click back on the project you just created. Assets should show 5 now.
3. Click ‘**Assets**’ to view all of the available assets that are in this project.

# **3: Define data source connections**

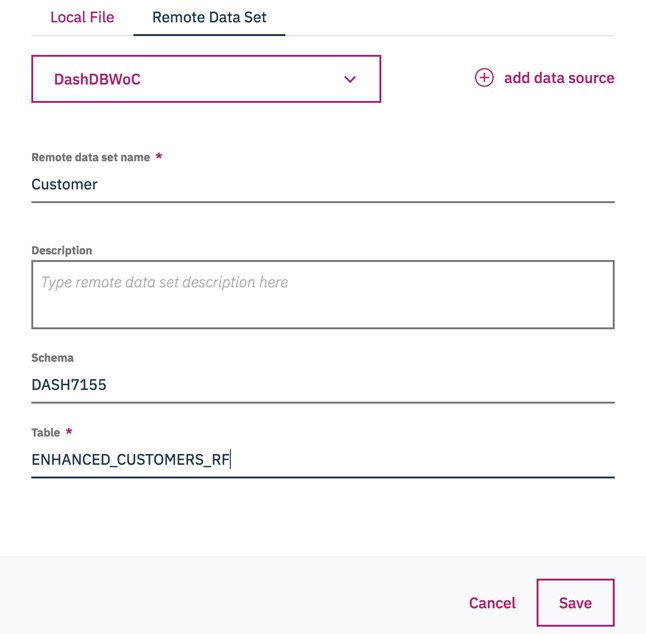
In this workshop, we want to make use of data from different sources. Notice that the project that you just imported has three local .CSV data files. Each of the three local files will be used in the labs. The Customer.csv and Churn.csv will be used in the Jupyter Notebook. The additional .csv will be used later as input data for testing the trained model. If it is available, we will attempt to also add the remote data source and connect to the merged data set that you created earlier in the Labs. For this lab, it is not critical to have the remote source available. But it will provide you knowledge of how to connect to remote data sources in the future.

**Step One:** **Create a data source connection** to a local DB2 Database:

1. From the project folder navigate to the ‘**data sets**’ section. You will notice that three CSV files were added to the project when you imported the project .zip file. Select ‘**add data set**’. Select the ‘**Remote Data Set**’ tab and click ‘**add data source**’ to define the data source connection.
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. **Data Source Name:** Provide a unique name that you will recognize for the class.
   2. **Data Source Type:** Select ‘Db2’
   3. **JDBC URL:** The JDBC URL that was provided earlier in class.
   4. **Username:** The Username provided earlier in class.
   5. **Password:** The Password provided earlier in class.
3. Click ‘**Create**’. The connection is created.

**Step Two:** 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 that you just added from the Pull-Down Menu.
2. Fill in the blanks with the corresponding data table information:
   1. **Remote data set name:** **Customer**.
   2. **Schema:** Will BE PROVIDED IN CLASS
   3. **Table**: Type in the TABLE NAME that you created containing the merged customer demographics and activity. If you followed the instructions in the earlier Lab, the name will be **ENHANCED\_CUSTOMERS\_{***YourInitials****}.*** If not, provide the table name that you used to create the table.



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 Jupyter notebook to test reading data from remote data sets and local files**

This will give you an introductory understanding of a Jupyter Notebook and how to load data into the 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 any name you would like.
   2. **Environment**: Select **‘Jupyter with Python3.5, Scala2.11, R 3.4.3 Spark 2.2.1**’
   3. **Language**: Select **Python 3.5**.
   4. Click ‘**Create’**
2. Next, we will generate the python code needed to import the data table from DB2 (or a remote database source).
   1. Click the blank in the first cell you see in the notebook.
   2. Click the '1001' 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 that we created earlier. **NOTE**: There is a known defect where “**Customers**” MAY not show up and will be blank.
   5. Choose the **'Insert Spark DataFrame in Python'** option. Note, the new Python code that is generated in the cell in your Notebook.
   6. Run the cell to show the data.