IBM Cloud Pak for Data

End to End AI Hands-On Lab

Telco Customer Churn

Table of Contents

Disclaimer

IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM's sole discretion. Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision.

The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract.

The development, release, and timing of any future features or functionality described for our products remains at our sole discretion I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results like those stated here.

Information in these presentations (including information relating to products that have not yet been announced by IBM) has been reviewed for accuracy as of the date of initial publication and could include unintentional technical or typographical errors. IBM shall have no responsibility to update this information. **This document is distributed "as is" without any warranty, either express or implied. In no event, shall IBM be liable for any damage arising from the use of this information, including but not limited to, loss of data, business interruption, loss of profit or loss of opportunity.** IBM products and services are warranted per the terms and conditions of the agreements under which they are provided.

IBM products are manufactured from new parts or new and used parts.
In some cases, a product may not be new and may have been previously installed. Regardless, our warranty terms apply."

Any statements regarding IBM's future direction, intent or product plans are subject to change or withdrawal without notice.

Performance data contained herein was generally obtained in controlled, isolated environments. Customer examples are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual performance, cost, savings or other results in other operating environments may vary.

References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business.

Workshops, sessions and associated materials may have been prepared by independent session speakers, and do not necessarily reflect the views of IBM. All materials and

discussions are provided for informational purposes only, and are neither intended to, nor shall constitute legal or other guidance or advice to any individual participant or their specific situation.

It is the customer's responsibility to insure its own compliance with legal requirements and to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer's business and any actions the customer may need to take to comply with such laws. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the customer follows any law.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products about this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products. IBM does not warrant the quality of any third-party products, or the ability of any such third-party products to interoperate with IBM's products. **IBM expressly disclaims all warranties, expressed or implied, including but not limited to, the implied warranties of merchantability and fitness for a purpose.**

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents, copyrights, trademarks or other intellectual property right.

IBM, the IBM logo, ibm.com and [names of other referenced IBM products and services used in the presentation] are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at: www.ibm.com/legal/copytrade.shtml.

U.S. Government Users Restricted Rights — use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM.

Introduction

In this lab you will learn to use Cloud Pak for Data for Telco Customer Churn problem which would in turn help you understand how to use various components of this platform to implement the lifecycle of this AI use case. This lab would cover

- Collecting data by connection to various data sources.
- Training and testing the machine learning (ML) model for predicting customer churn.
- Deploying the ML model as a web service
- Monitoring the ML model at runtime for fairness and explainability.

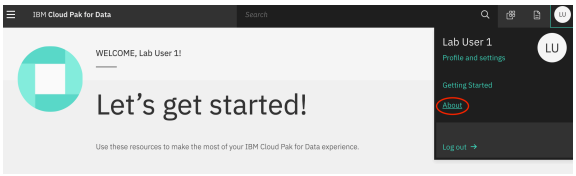
Cloud Pak for Data supports multiple approaches for collecting the data for modeling, including:

1. *Getting the data for modeling from a remote data store (such as DB2Wh on Cloud) directly by using Data Virtualization. If your Cloud Pak For Data cluster has Data Virtualization enabled, you can use this approach.*
2. *Reading the data from local csv files.*

In this lab, we will show code how to apply both approaches for data collection. Due to time constraints of the lab, we'll leverage the approach of reading data from a local csv file. We will include the instructions for Data Virtualization in the Appendix.

Please note that the figures in this lab were obtained from ICP4D version 2.1.0.1. Different version may have slight variations in the UI but the functionality outlined in this hands-on lab will be supported. For latest updates, always check the documentation.

To check the version of your ICP4D cluster, click on the circle in the top right which has your username initials and press About.



1. Pre-Requisites

This hands-on lab illustrates how to train, test, and deploy machine learning models for customer churn prediction in a hybrid cloud environment. It shows how to collect data and train the model on IBM Cloud Pak for Data environment which is already setup for you and then how to monitor that model using Watson Openscale on IBM Public Cloud. You will need to setup and access an IBM Public Cloud account to be able to complete this lab.

Please make sure to run through all the steps in the pre-requisites document before starting the lab (https://ibm.biz/dataaiprereqs)

2. Access Cloud Pak for Data platform

Your instructor will provide you with access credentials to a Cloud Pak for Data cluster which is already installed and setup for you to execute the steps in this lab.

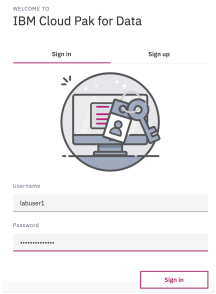
Cloud Pak for Data Credentials:

url = '####'
username = '####'
password = '####'

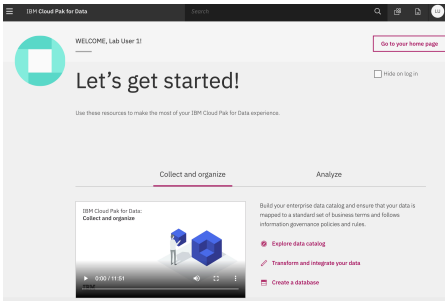
Cloud Pak for Data supports multiple roles including 'Business Analyst', 'Data Engineer', 'Data Scientist', 'Data Steward', and 'Developer' with each role having access to the tools and data as defined by that role (for more details, check this link).

The username and password provided for this lab have access as 'Data Scientist', 'Data Engineer' and 'Developer' role which are needed for various steps in this lab.

Use the provided credentials (url, username, and password) to login.



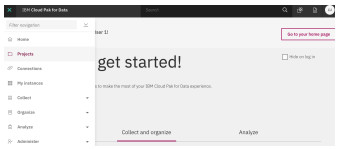
On successful login, you will see following ICP4D landing page.



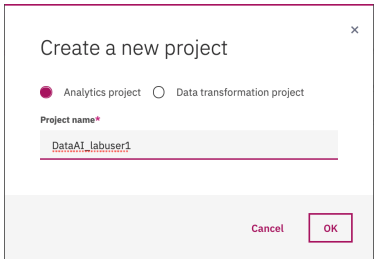
3. Create Analytics Project

In IBM Cloud Pak for data, a project is how you organize your resources to achieve a particular goal. A project allows for high-level isolation, enabling users to package their project assets independently for different use cases or departments. Your project resources can include data, collaborators, scripts, and analytic assets like notebooks and models.

1- Click on the navigation menu ("3 horizontal bars" or hamburger icon) on the top left and select **Projects** from the drop down.



2- Click on **New project** to create a new project.



In IBM Cloud Pak for Data, there are 2 major types of projects that you can create:

- Analytics project refers to data science related projects
- Data transformation project refers to Extract/Transform/Load (ETL) related projects

An analytics project is a collection of assets that you use to achieve a particular data analysis goal. Your project assets can include:

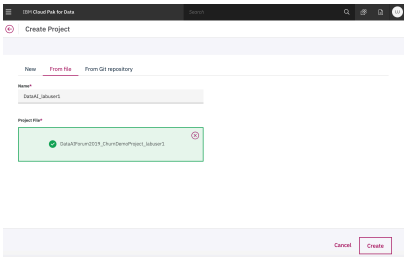
- Notebooks
- RStudio files
- Models
- Data assets (local files, data sources, and remote data sets)
- Scripts

Select **'Analytics project'** and provide a unique Project name (**DataAI_yourusername**). Your **username** would be what the instructors provided you.

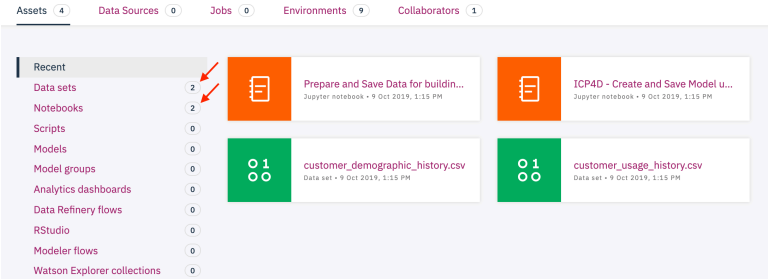
Press OK.

3- Download the sample project file DataAIForum2019_handsonlab_project.zip from github.
Project file: https://ibm.biz/cp4dproj
After download, rename downloaded file to DataAIForum2019_ChurnDemoProject_**<yourusername>**.zip.

To **import** project select **'From file'**. Drag and drop or browse for the zip file you downloaded (DataAIForum2019_ChurnDemoProject_**<yourusername>**.zip).



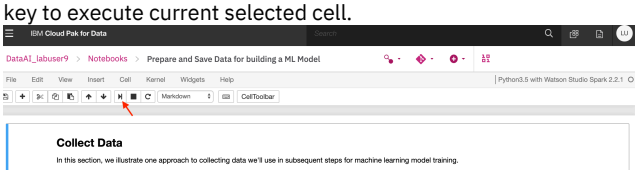
4- Click '**Create**', then the project is created. You may have to click in in Name field and tab to next field in order to see 'Create' button. The imported project has "Assets" including 2 Data Sets and 2 notebooks that will be available for this project.



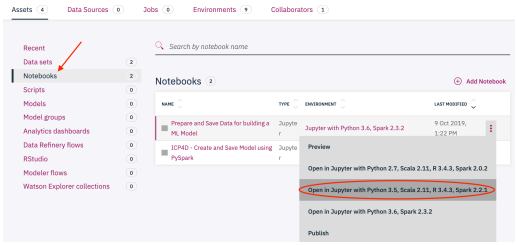
4. Build ML Model using Jupyter Notebook

In this section, we will train a machine learning model in a Jupyter notebook using the data assets in our project.

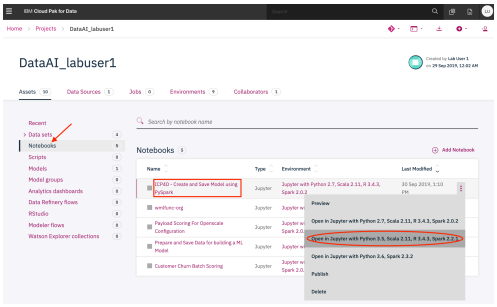
Jupyter Notebook is an interactive exploration and development tool used by data analysts, data engineers and data scientists. We will use the notebooks "Prepare and Save Data for building a ML Model" and "ICP4D-Create and Save Model using PySpark" that have been imported as part of our project. The notebooks contain comments which will guide you through their use. As you review each cell click in it with your mouse and use the run button on the toolbar (annotated with red arrow in figure below) to execute it (short-cut is to use **Shift-Enter** key to execute current selected cell.



1- Navigate to the project you created earlier by click the Home menu ➔ **Projects** and selecting the project you created earlier.
2- Click **Notebooks** tab and open "Prepare and Save Data for building a ML Model" notebook using Jupyter with Python 3.5 with Spark 2.2.1 environment. **You can select this environment by clicking the 3 vertical dots at the right of the name of the Notebook.**



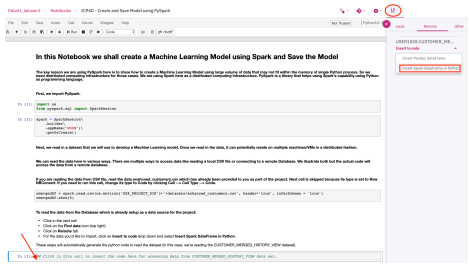
3- Follow the notebook instructions and execute all cells in the notebook.
4- Once you're done executing the "Prepare and Save Data for building a ML Model" notebook, navigate back to **Notebooks** tab and open the "ICP4D-Create and Save Model using PySpark" notebook using Jupyter with Python 3.5 with Spark 2.2.1 environment.



5- Follow the notebook instructions and execute all cells.

The notebook includes the instructions but it is worthwhile mentioning here for reference how to access the datasets to be used by the notebooks:

- Click in the empty cell (only has a comment) as shown in the figure below (annotated with red arrow)
- Click the **Find data** icon (annotated with red oval in figure below)
- Click Insert to code and select Insert Spark DataFrame in Python as shown in the figure below (annotated with red rectangle).



NOTE : Make sure to select Spark DataFrame option and also modify variable "**df**" (generated code use df2,df3, df4, ... based on number of attempts). Rename the generated dataframe as explained in the notebooks.

6- Follow directions in the Notebook to execute every cell.
In the cell similar to the image given below you will save the model into the project repository for further use. Please ensure that when you are saving the model with a unique name (include your username in the model name, for example specify the model name as "<yourusername> Telco churn prediction model").

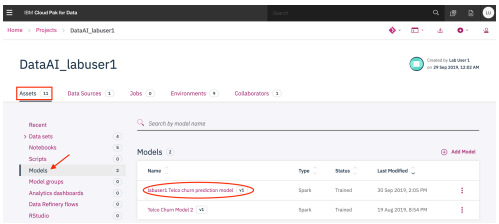
Now we save the Model. Please note the URL that is created after teh model is saved.

```
In [22]: from dsx_ml.ml import save

In [23]: model_name = "labuser1 Telco churn prediction model"

In [ ]: save(name = model_name, model = model, algorithm_type = "Classification", test_data = test)
```

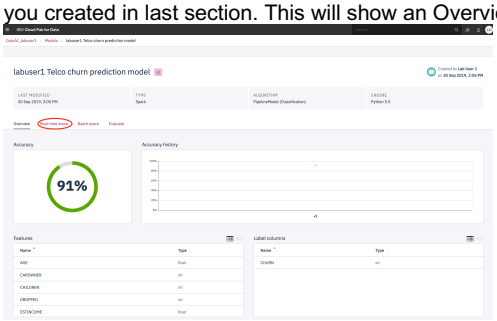
7- Verify that the model is saved under the model section of the project by clicking on the project name (DataAI_<yourusername>) and clicking the **Assets** tab. Look under the **Models** tab (annotated with red arrow) to make sure that the model is shown. Your model will have a different name and version.



5. Test and Validate ML Model

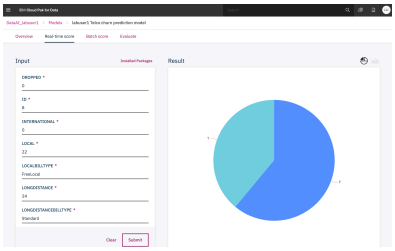
In this section, we test the ML model we trained by creating a sample single case scoring scenario.

1- Navigate to the Models tab in the **Assets** view of your project. Click the model name that you created in last section. This will show an Overview of the model asset.



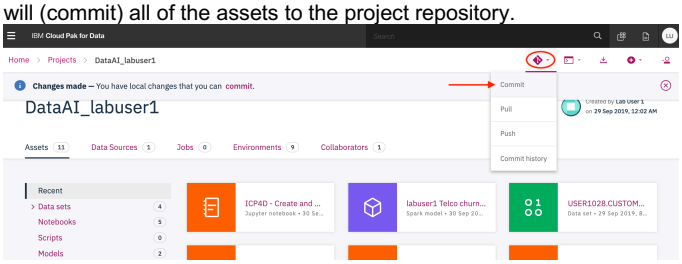
2- Select the **Real-time score** tab(annotated with red oval in figure above). Sample data is automatically generated and displayed in the “Input” section for each of the fields (features) in the model.

3- Click **Submit** and the online score will show on the right side “Result” section. You can change some of the input values to see how it changes the prediction.

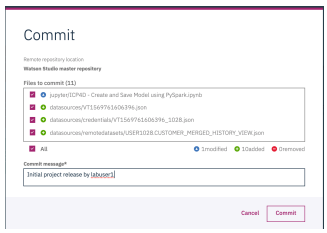


Next we will prepare the model artifacts for deployment in production setup so that we can score new records. The data scientists may tag a model once satisfied with the result as “published” for production consumption. In Cloud Pak for Data platform this step is called creating “**Project release**”. To do so, we need to commit changes and push them to git repository.

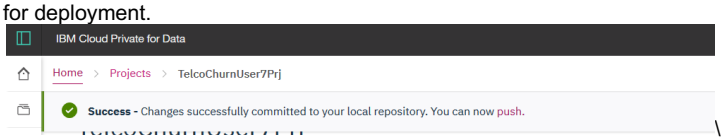
4- Navigate to the project homepage by clicking the Project Name. Click on **Git actions** and click **Commit** from top right menu (similar to that shown in the figure below). This will (commit) all of the assets to the project repository.



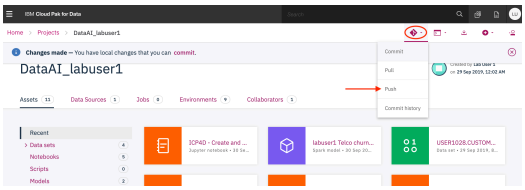
5- The “commit” screen will show list of the assets that are created in this project (similar to that shown in the diagram below). Enter a suitable comment (for e.g. “This is initial project release by labuser1”. Click **Commit**.



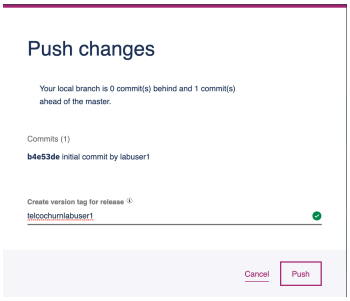
You will see a message window confirming the changes were commited and now available to “push” for deployment.



6- Click on **Git Actions** and then click **Push** (as shown in figure below).



7- Provide a 'tag' for your release so that a version of this model can be identified for deployment (for ex, enter "telcochurn<yourusername>").
8- Click **Push**.

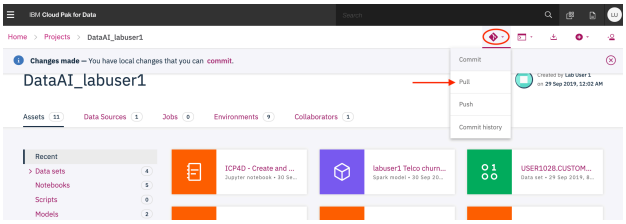


You should see a confirmation message window indicated changes were pushed successfully.

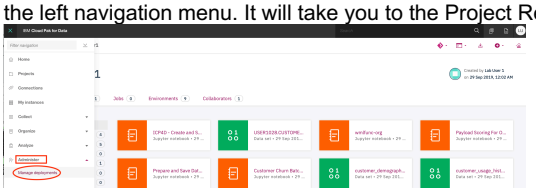
6. Deploy ML Model using Project Release

At this point, the model can be deployed to production. To do so, we need to first create a release for this project. A project release represents a project tag that can be launched as a production environment. Project releases and deployments can be monitored from the Project releases page. Note that the user deploying the model can be different from the user who trained and tested the model (in this lab, we're assuming the same user does both).

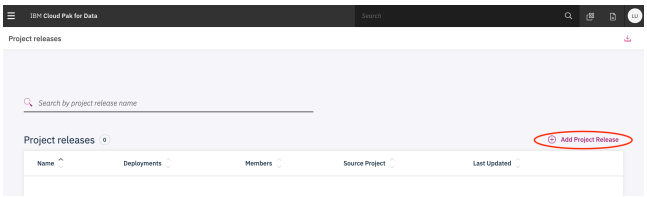
1- Navigate to the project you've been working on by clicking on Home Menu and then **Projects**. Then select your project (DataAI_<yourusername>)
2- Click **Git Actions** and then click **Pull**. You may get the message saying you are up to date (which means that you have the latest version of the project updates).



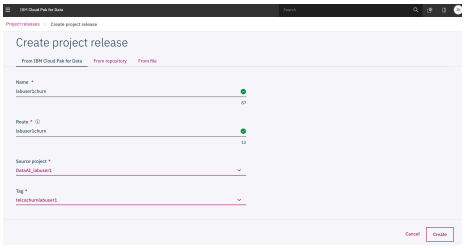
3- Navigate to the home menu and select **Manage Deployments** under **Administer** from the left navigation menu. It will take you to the Project Release page.



4- To add a new project release, click **Add Project Release** from right handside option.



5- Add in the details of your Project Release under the '**From IBM Cloud Pak for Data**' tab and click **Create**.



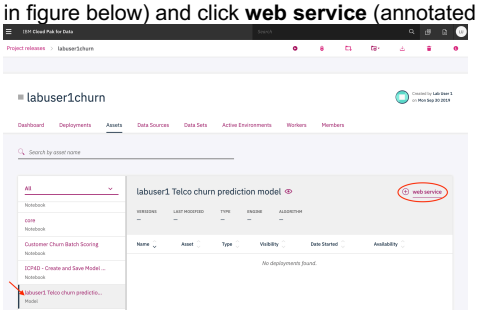
These details should be unique. For e.g.

Name : <yourusername>churn
Route : <yourusername>churn
Source Project : <your_project_name> (This is the project you are working so far on)
Tag : <your_project_release_tag> (this is the tag you entered when pushing the project to Git repository)

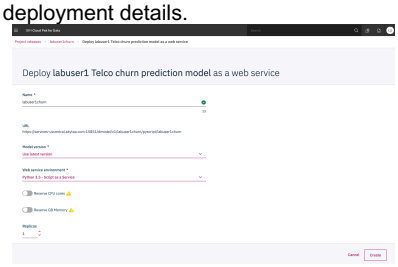
The name can contain hyphens but not special characters such as a period (.). The route is the unique ID for the project release, and is used within the deployments' REST paths and URLs. This is a unique part of the url that will be assigned to all of the assets that are created related with this project. Name of the “route” must be lowercase.
IMPORTANT Please remember the route you provided in this step.

Next, we will create a realtime online deployment of the model. The ICP for Data platform will package the required model artifacts as a docker container along with any runtime execution environment. This container is an isolated image from other “development” services that are running. An administrator can provison multiple replicas of this service in order to fulfill concurrent access to model scoring in production.

6- In the **Assets** tab (annotated with red rectangle in figure below), you can see all of the analytics assets. There are notebooks, models, and scripts that we created previously. Select the model with the name that you created in model creation step, (say something like **‘<yourusername> Telco churn predition model)** on the left (pointed to by arrow in figure below) and click **web service** (annotated with red oval in figure below).



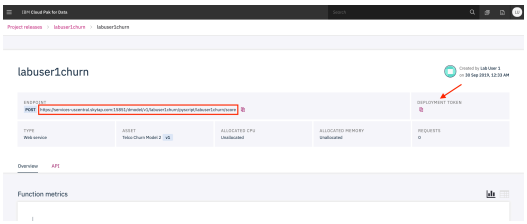
7- This will load the “Deploy” window expecting input as shown below. Fill in the deployment details.



For example :
Name : <yourusername>churn (it should contain only lower case letter and numbers)
Model version : Use latest version
Web service environment : Python 3.5 – Script as Service. Please note that programing language and its version of the environement you select here should be the same as the ones

you have used for creating the model. For example, if you have used **Python 3.5 Spark 2.3.2** as environment while developing the model here the environment you select should be **Python 3.5 – Script as Service**

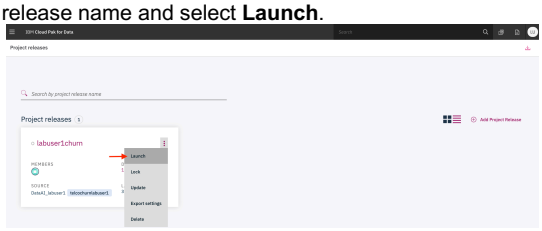
8- Click **Create**. This will load the deployment Overview screen with API tab as shown below.



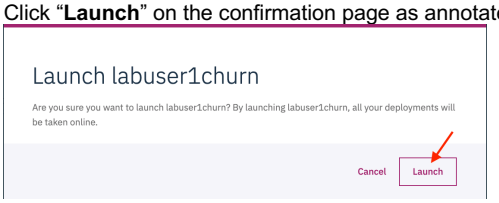
Note your Web service ENDPOINT URL (annotated with red rectangle in figure above) and Deployment Token (annotated with red arrow in figure above). Copy these as you will need them in Setting up Model Monitoring step.

At this time, the online deployment is created. However, note that the deployment is not active as yet. We will need to "Launch" the deployment in order instantiate required execution environments.

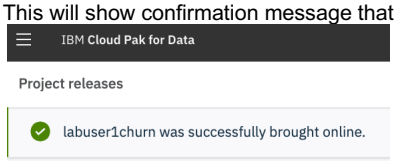
9- Click on **Project releases** tab from top left side tab. Click on 3 dots right of project release name and select **Launch**.



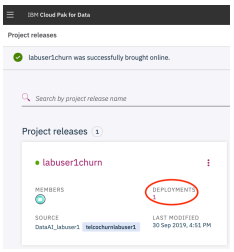
10- Click "**Launch**" on the confirmation page as annotated in the figure below with red arrow.



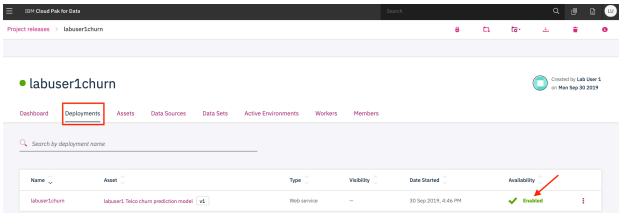
This will show confirmation message that deployment is successfully brought online.



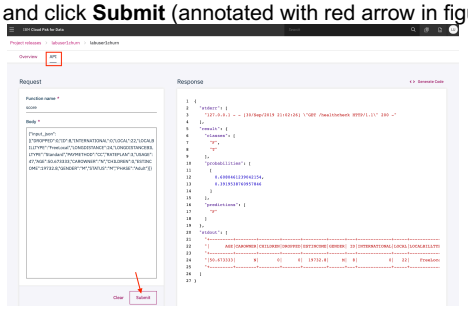
Actual instantiantion of environments and setup could take 1-2 minutes before APIs can be accessible. Under project releases , “Deployments” will show count as 1 as annotated with red oval in the figure below.



11- Click on Project release name to load back the Dashboard. Click on **Deployments** tab (annotated with red rectangle in figure below) which will show 1 model with “Avaliablity” as “Enabled” . (The enabled status may not change for 1-2 minutes after “**Launch**” step so please allow it some time).



12- Click on deployment name (for e.g. ***<yourusername>churn***) to bring up “Overview” and “API” tabs for this deployment. To test the endpoint is functioning, we can run a sample record to score. Click on **API** tab (annotated with red rectange in figure below) and click **Submit** (annotated with red arrow in figure below) under “Body” window.



This will show scoring of 1 record using real-time deployment of the model. The Response is shown as JSON record. You are here mimicing the invokation of the Rest API for scoring the model as you would do to for online real time Scoring.

Note: Generate Code option on top right shows how to invoke REST API call using cURL command.

For a quick recap, we've leveraged Cloud Pak for Data platform to train a churn prediction random forest model using Jupyter notebooks and Spark, and deployed the trained model as a web service accessible via REST API.

7. Setting up Model Monitoring

In this section, we will illustrate how to leverage Watson OpenScale on IBM Public Cloud to monitor the trained and deployed machine learning model for Telco churn prediction on this Cloud Pak for Data instance.

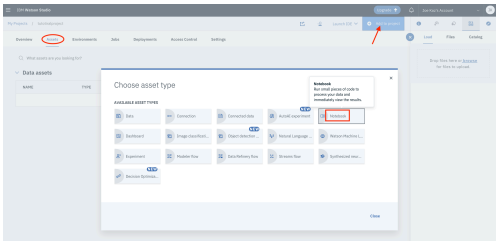
IBM® Watson OpenScale tracks and measures outcomes from your AI models, and helps ensure they remain fair, explainable and compliant wherever your models were built or are running. Watson OpenScale also detects and helps correct the drift in accuracy when an AI model is in production

Watson Openscale can monitor machine learning models deployed by several machine learning service providers including Watson Machine Learning, Amazon Sagemaker, Microsoft Azure ML Studio and Microsoft Azure ML Service. You can also configure a "Custom environment" for monitoring.

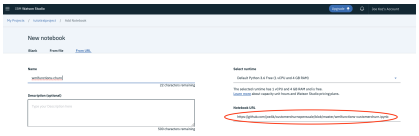
7.1 WML Function Wrapper

The model we have trained and deployed doesn't match any of these service providers so in order to monitor the deployed model with Watson Openscale, we will build a wrapper function and deploy it to Watson Machine Learning.

1- Launch Watson Studio by logging into your IBM Cloud account, finding the Watson Studio service you had created earlier as part of the pre-requisites and clicking **Get Started** (the pre-requisites document offers details on this step, https://ibm.biz/dataaiprereqs).
2- Click on the project you had created as part of the pre-requisites (or you can create a new project if you wish).
3- On the project page, click **Assets** tab (annotated with red oval), click **Add to project** (annotated with red arrow) and select **Notebook** (annotated with red rectangle) as show in the figure below.



4- Click on the From URL tab, provide a name for your notebook, select the Python 3.6 Free runtime environment and paste the following URL in the Notebook URL field.
https://ibm.biz/wmlfunc
(long url in case short url doesn't work,
https://github.com/joe4k/customerchurnopenscale/blob/master/wmlfunctions-customerchurn.ipynb)

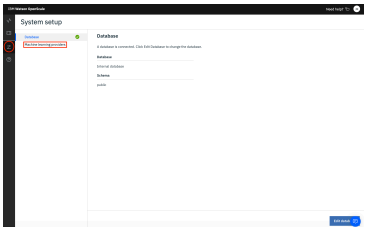


5- Read the instructions and execute the cells in the notebook.
The main edits/changes you need to make include:
   - Providing your Watson Machine Learning service credentials. In the cell with wml_credentials, provide the "apikey", "instance_id", and "url" information from the service credentials of Watson Machine Learning service you created (https://ibm.biz/dataaiprereqs includes more details).
   - Providing the cp4d_scoring_url and auth_token in the function called "churn_deployable_function". These values you copied in "Deploy ML Model using Project Release" step above and they're effectively the REST endpoint and authorization tokens to access the machine learning model you had previously deployed.
   - Providing a unique name for your deployed function to WML. It is recommended to include your username, something like "Customer Churn ICP4D <yourusername>").

7.2 Watson Openscale Setup

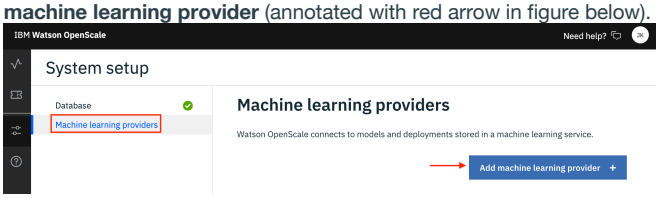
1- Launch Watson Openscale by logging into your IBM Cloud account, finding the Watson Openscale service, and clicking **Launch Application** (the pre-requisites document offers details on this step, https://ibm.biz/dataaiprereqs).

2- Select the configure tab (3rd symbol down from left navigation column, annotated with red circle in figure below).

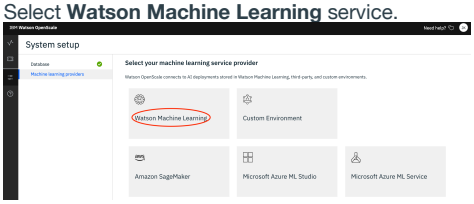


3- Keep the Database selection the default which is an internal database. This is useful for demonstration purposes but not for production. For production, you will need to create a separate database (Db2 or PostgreSQL). Openscale needs the database to log scoring responses which would then be used for fairness, accuracy, and explainability.

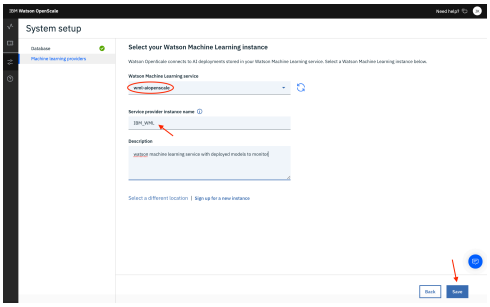
4- Click on Machine learning providers (annotated with red rectangle in the figure below) to configure the machine learning service where models are deployed. Then click on **Add machine learning provider** (annotated with red arrow in figure below).



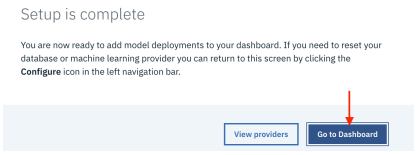
5- Select **Watson Machine Learning** service.



6- On the next screen, select the Watson Machine Learning service where you've deployed the model (annotated with red oval in figure below). Provide an instance name for you to identify this machine learning service and click **Save**.

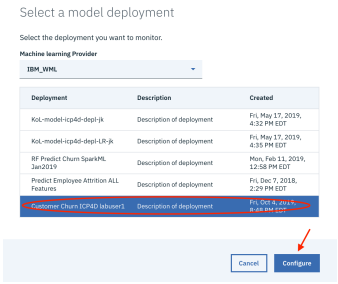


7- You should see a "Setup Complete" message as shown below. Click on Go to Dashboard.

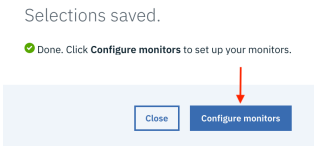


8- On the next screen, click **Add** to add the deployed model to monitor. This will pop-up a window for you to select from all the models deployed on that Watson Machine Learning service.

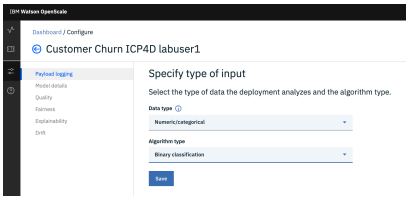
9- Select the model you've deployed in WML Function Wrapper step (7.1) above and click **Configure** (annotated with red arrow in figure below). Note that the list of models you will see may be different from what is in the figure below.



10- You should see a "Selections saved" message as shown below. Click **Configure monitors**.

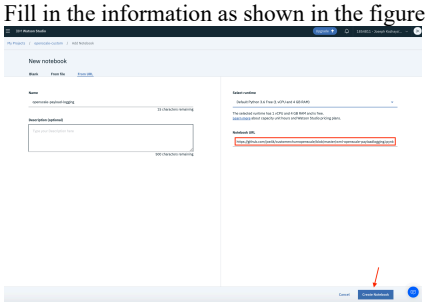


11- Provide the information for "Payload logging" as shown in figure below. Click **Save**.



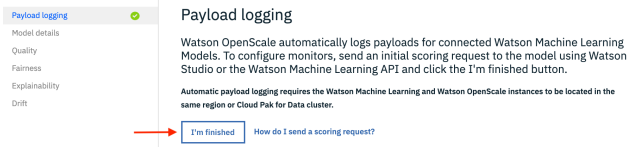
12- The next screen explains that Watson Openscale needs one initial scoring request to be executed to initiate payload logging. To do so, please execute the following notebook in your Watson Studio project.

a. Navigate back (or Launch) Watson Studio as described earlier.
b. Navigate to the project you've been working with or create a new project.
c. Under Assets tab, click New Notebook.
d. Select from URL tab.
e. Specify the Notebook URL:
https://ibm.biz/payloadlogging
(long url in case short url doesn't work, https://github.com/joe4k/customerchurnopenscale/blob/master/wml-openscale-payloadlogging.ipynb)
f. Fill in the information as shown in the figure below and click **Create Notebook**.

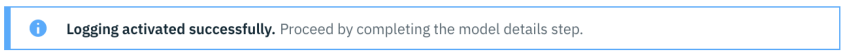


g. Execute the steps in the notebook. You mainly need to provide the credentials to your Watson Machine Learning service and then specify the deployment id for the model you're monitoring.

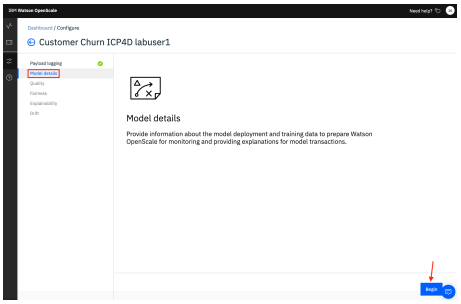
13- Back in Openscale UI, click on **I'm finished** (annotated with red arrow in figure below).



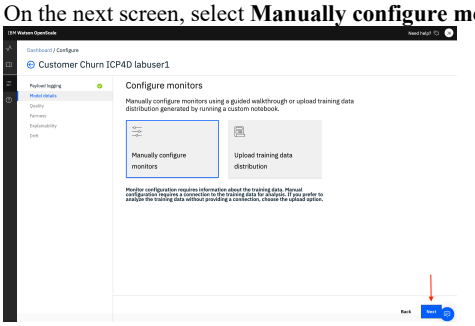
14- You will see a confirmation message indicating that logging was activated successfully.



15- After payload logging setup is complete, click on "Model details" and click **Begin**.

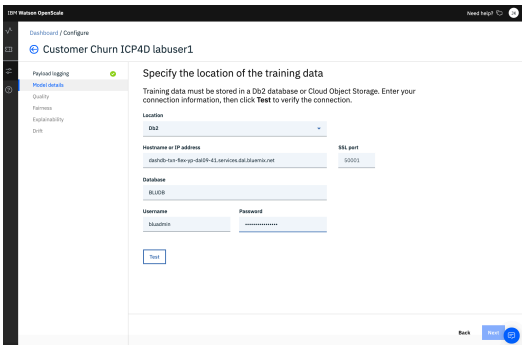


16- On the next screen, select **Manually configure monitors** and click **Next**.



17- On the next screen, provide information to access the database for training data.
Location: Db2
Hostname or IP address: ##############
SSL Port: #########
Database: #########
Username: ##########
Password: ##########

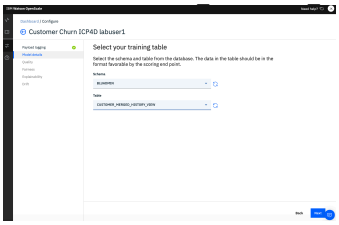
Deleted: dashdb-txn-flex-yp-dal09-41.services.dal.bluemix.net
Deleted: 50001
Deleted: BLUDB
Deleted: dataforum2019
Deleted: yJV7tz8O%G3syJV7tz8O%G3s



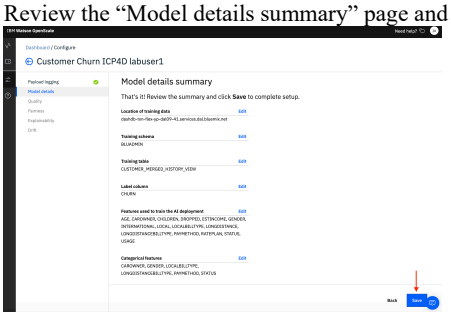
Click **Test** and once you get a message for successful connection, click **Next**.

18- On the next screen, select the following Schema and Table with training data.

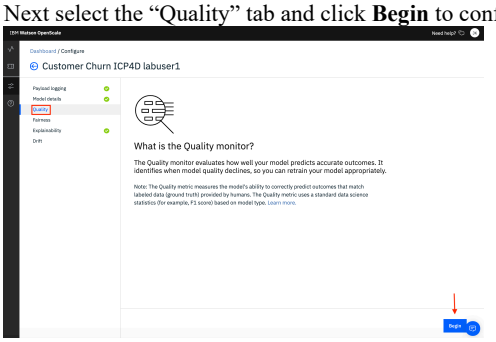
Schema: BLUADMIN
Table: CUSTOMER_MERGED_HISTORY_VIEW



19- Next, select CHURN as the label column from training data and click **Next**.
20- Next, select the features used to train the model. Keep all features selected except for ID field and click **Next**.
21- Next, select text and categorical features. Do not make any changes (keep selected features). Click **Next**.
22- Next, select deployment prediction column. Select "prediction" field and click **Next**.
23- Next, select probability as the feature that contains confidence value for each prediction and click **Next**.
24- Skip the next step (transaction ID) and click Next.
25- Review the "Model details summary" page and click **Save**.



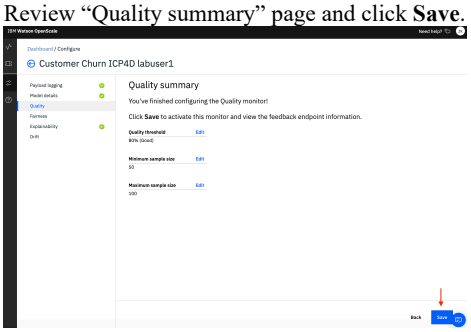
26- Next select the "Quality" tab and click **Begin** to configure quality monitor in Openscale.



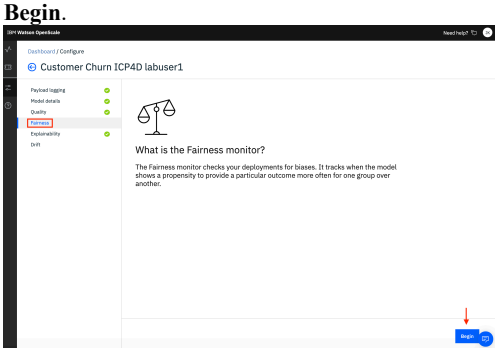
27- Set quality alert threshold by keeping the default of 80% and click **Next**.

28- Next, set minimum sample size to 20 and maximum sample size to 100. In production, you should use larger sample sizes to make they're representative of the requests the model receives. Click **Next**.

29- Review "Quality summary" page and click **Save**.



30- Next, select "Fairness" tab (annotated with red rectangle in figure below) and click **Begin**.



31- Next, on "Specify the favourable outcomes" page, specify **F** (false) as Favorable value and **T** (true) as Unfavorable value. Click **Next**.

32- Next on "Select features to monitor" page, select GENDER and AGE features to monitor and click **Next**. These would be the features to be monitored for fairness.

33- Next on "Specify reference and monitored groups [GENDER]" page, specify **M** and click **Add** to add **M** to "Reference group" and specify **F** and click **Add** to add **F** to "Monitored group".

34- Next on "Set fairness alert threshold [GENDER]" page, select 80% as the alert threshold and click **Next**.

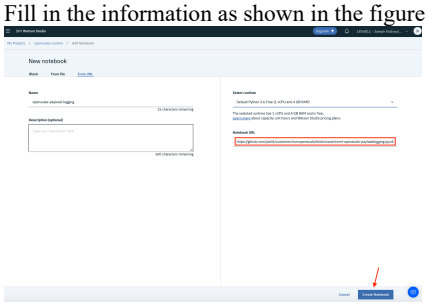
35- Next on "Specify reference and monitored groups [AGE]" page, keep the recommended selections and click **Next**. If Watson Openscale does not automatically recommend age ranges for reference and monitored groups, specify the following

- Reference Group: 38.4 (Starting value) – 52.8 (End value)
- Monitored Groups:
  12.3 (Starting value) – 23.3 (End value)
  23.3 (Starting value) – 38.3 (End value)
  52.9 (Starting value) – 77.1 (End value)

36- Next on “Set fairness alert threshold [AGE]” page, select 80% as the alert threshold and click **Next**.
37- Next on “Set minimum sample size”, specify 25 and click **Next**.
38- Review the configuration information on the “Fairness summary” page and click **Save**.

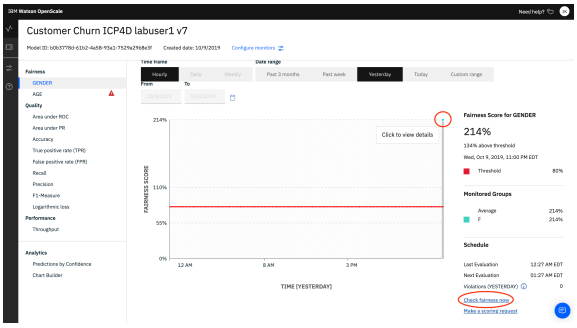
You have successfully completed Watson Openscale configuration for fairness, accuracy, and explainability to monitor the customer churn prediction machine learning model.

In production, as your machine learning model is accessed by applications, Watson Openscale will monitor those scoring events and provide a dashboard (as well as APIs) that business/AI Ops users can leverage to detect undesirable behavior and establish trust in the AI models.
To emulate that behavior, the last step in this hands-on lab is to execute batch scoring notebook so Openscale can collect enough data points to report some results.
39- Execute batch scoring notebook in Watson Studio.
   a. Navigate back (or Launch) Watson Studio as described earlier.
   b. Navigate to the project you’ve been working with or create a new project.
   c. Under Assets tab, click New Notebook.
   d. Select from URL tab.
   e. Specify the Notebook URL:
      https://ibm.biz/openscalebatchscoring
      (long url in case the short url doesn’t work for some reason,
      https://github.com/joe4k/customerchurnopenscale/blob/master/customerchurn_batch_scoring.ipynb)
   f. Fill in the information as shown in the figure below and click **Create Notebook**.

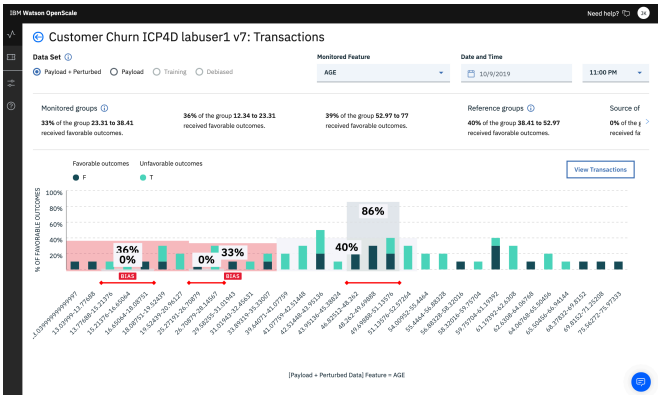


   g. Execute the steps in the notebook. You mainly need to provide the credentials to your Watson Machine Learning service and then specify the deployment id for the model you’re monitoring.

40- Once the notebook finishes execution successfully, navigate back to Watson Openscale and click on Check fairness now (annotated with red oval in figure below).



41- This should show the fairness metric on the dashboard and you can click on that specific metric (little green dot annotated with red circle in figure above) to get a more detailed view.



Conclusion

Congratulations on completing this end-to-end hands-on lab on collecting data, training, testing, and deploying a machine learning model using Jupyter notebooks in IBM Cloud Pak for Data and monitoring the deployed model for accuracy, fairness, and explainability using Watson Openscale on IBM Public Cloud.

Appendix

Data Virtualization

Create Virtualized Data Assets

One of the first tasks when using Cloud Pak for Data platform is to connect to the data sources that will feed the analytics projects of the enterprise. Cloud Pak for Data offers the differentiating capability of Data Virtualization which integrates data sources across multiple types and locations into one logical data view. This virtual data platform enables real-time analytics without moving data, duplication, ETLs, and additional storage requirements, so processing times are greatly accelerated.

In this section we will perform full lifecycle of data virtualization process.

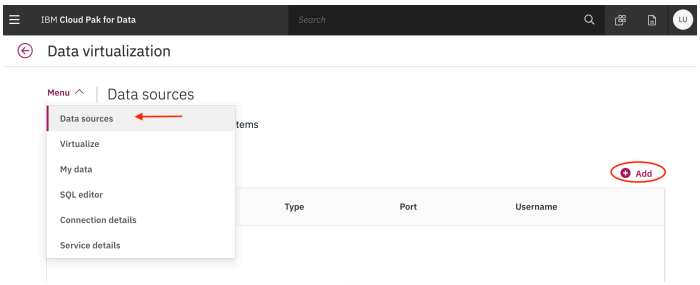
Manage Users

The first step in Data Virtualization is to manage users and provide access to Data Virtualization capabilities to your ICP4D users. **This step is performed only once and it has been done in your environment already. For guidelines on how to apply this step to your own environment in the future, please check the details in the documentation.**

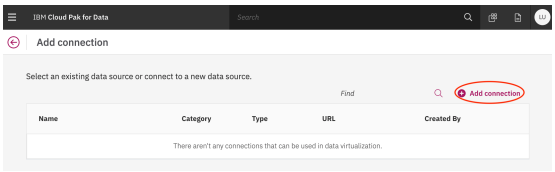
Add data sources

Data Virtualization supports many relational and non-relational data sources that you can add to your data source ecosystem. Data Virtualization agents connect to relational data sources using the Java Database Connectivity (JDBC) protocol. The credentials used to establish the connection to the data source determine what data in the data source can be accessed by the Data Virtualization agent.

1- Click on Home menu "**Collect**" option, then click on "**Virtualized data**". On the Data virtualization page, click on "**Menu**" drop down and select "**Data sources**".
If the DB2Warehouse connection doesn't appear, click on **Add** and then select **Add data source.** Note that in a multi-user environment, once the data source connection is setup, it becomes available to other users on the platform assuming their roles allow them access to such data sources.



2- Click on **Add connection** which will bring up a new window to provide connection details.



3- Fill in the connection details provided by the instructors. Specifically, you need the DB2
Connection name: db2wh_cloud_<yourusername>
Description: *optional* add a description for this data connection.
Connection type: Db2
Host: ########
Port: ########
Database: ######
Username: ########
Password: ############

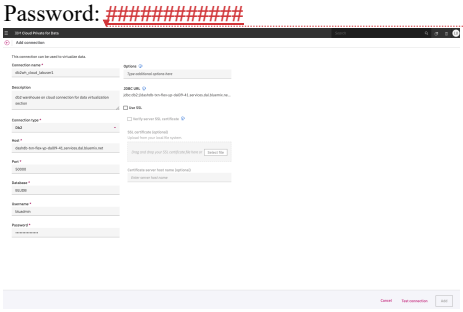
Deleted: dashdb-txn-flex-yp-dal09-41.services.dal.bluemix.net...

Deleted: 50000

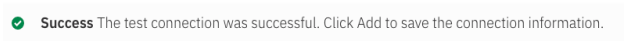
Deleted: BLUDB

Deleted: dataforum2019

Deleted: yJV7tz8O%G3syJV7tz8O%G3s



After providing all parameter values, click on **Test Connection**. It should return a message indicating Success which mean the credentials are valid and the data source can be connected to successfully.



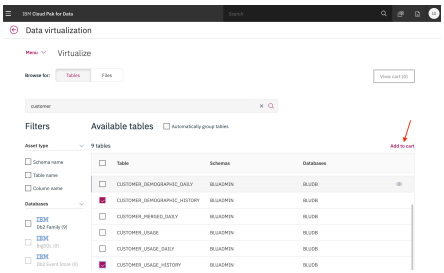
Once you get this message, click on **Add** to add this connection to Data Virtualization on your ICP4D cluster.

Create virtual assets

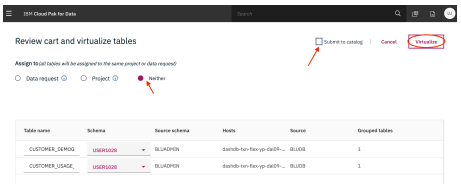
The most common mechanism for virtualizing data is to create a table "view" or virtual table. You can create a virtual table to segment or combine data from one or more tables. Tables that are similar from multiple sources can be combined into a single virtual table, which will create a unified definition containing the columns and data from all participating data sources. Segmentation can be vertical (either a subset or superset of columns based on a selection of chosen columns) or horizontal (an explicit set of rows or records based on a conditional

expression) or both. You can then run queries against the resulting virtual table no different than how you would query any of the base tables.

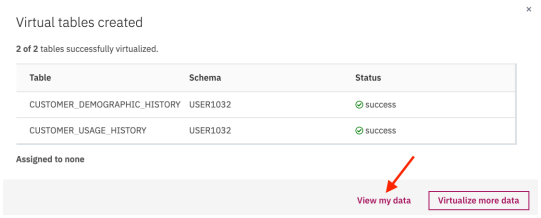
1- If you are not in the Data Virtualization UI, click on ICP4D Menu (top left) and click on **Collect** and then **Virtualized data**.
Once in Data Virtualization UI, click on **Menu** and then select **Virtualize**.
2- Search for customer in the Search Box. That will show the tables that contain customer information from the Db2 Warehouse on Cloud connection we created earlier.



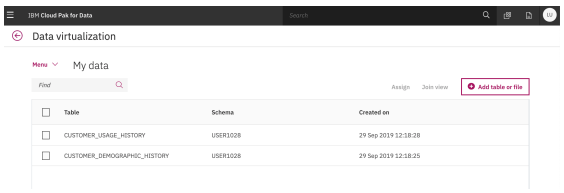
3- Select the CUSTOMER_DEMOGRAHIC_HISTORY and CUSTOMER_USAGE_HISTORY tables.
4- Click on **Add to cart** (on the right above list of tables as shown with the arrow in the above figure) and then click on **View cart (2)**. This will show selected tables in your shopping cart.
5- Select **Neither** under **Assign to** option.
6- Uncheck the Submit to catalog option.
7- Click **Virtualize**.



8- On the results window, click **View my data**. This will redirect you to view all data you have so far virtualized.



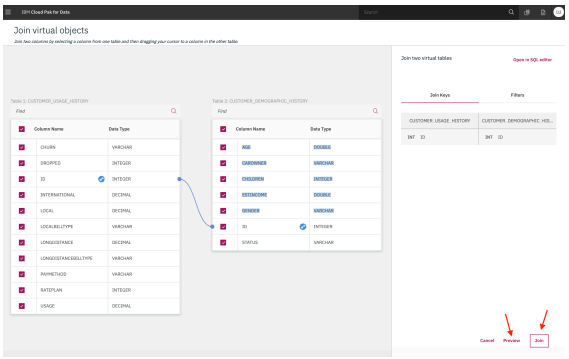
9- In the My data view, you can see the two tables you've just added to your data assets.



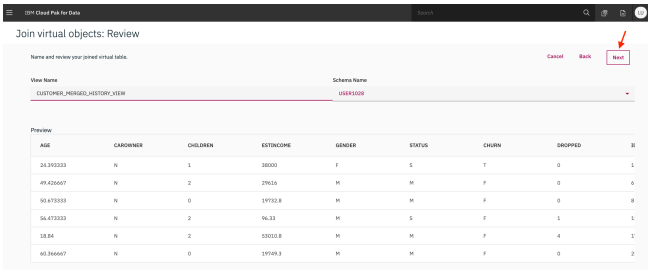
Create a joined view

1- If you are not in the Data Virtualization UI, click on Home menu **"Collect"** option, then click on **"Virtualized data"** . Once in the Data Virtualization UI, click on **"Menu"** and then on **"My data"**. Select CUSTOMER_DEMOGRAHIC_HISTORY and CUSTOMER_USAGE_HISTORY. (if you're continuing from previous step, you will already be on the My data view and you can proceed to step 2)

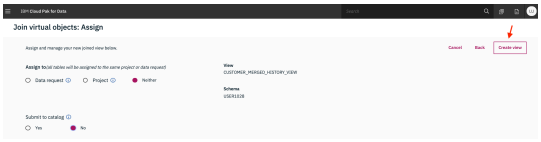
2- Click **Join view**. This will bring up both tables schema side-by-side with option to drag connection to identify joining column/s. Click in the ID column area of CUSTOMER_DEMOGRAHIC_HISTORY table and drag while holding left mouse button and bring it over ID column on CUSTOMER_USAGE_HISTORY table area.



3- Click **Preview** to see that the columns from CUSTOMER_DEMOGRAPHIC_HISTORY show up with the columns from CUSTOMER_USAGE_HISTORY.
4- Click **Join** which shows a "Join virtual objects: Review" screen where you can enter the name for the new joined virtual table. Enter in the name CUSTOMER_MERGED_HISTORY_VIEW.



5- Click **Next** which shows the "Join virtual objects:Assign" screen. Under "Assign to", select "Neither" option. Under "Submit to catalog" option, select "No". Finally click **Create view**. This will show a confirmation screen showing the join has been created. Click **View my data**, to verify.



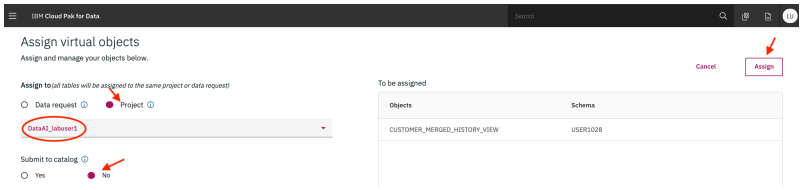
To quickly recap, this step shows how to leverage Data Virtualization capability of IBM Cloud Pak for Data platform to connect to a Db2 Warehouse on IBM Public Cloud, select two tables, and create a virtualized joined view of those two tables. You can use the virtualized data in several different ways. For example, you can use them in a Jupyter notebook, create new models within the Model Builder, or build charts or graphics on the analytics dashboard.

Assign assets to a project

After you create an Analytics Project, you assign the virtualized joined table created earlier to this project.

1- Click on Home menu **Collect** option, then click on **Virtualized data**. Once in the Data Virtualization UI, click on **Menu** and then on **My data**. Select only CUSTOMER_MERGED_HISTORY_VIEW and click **Assign**.

2- On the "Assign virtual objects" screen, select Project radio button and select the name of the project that was created in last section (DataAI_<yourusername>). Select "No" for Submit to catalog option and click **Assign**.



3- On the confirmation screen click on **Go to project** to verify. Notice that the project now has 4 Data sets (originally when we imported it, it had only 3 Data sets).