

IBM Journey to Cloud and AI

Analytics Modernization Workshop

Featuring: Cloud Pak for Data (CPD)

Lab Workbook



Lab workbook & CPD environment by: **Burt Vialpando**, Executive Analytics Architect

Adapted from original work by: **Vikram Khatri**, Executive Architect, Cloud Pak for Data (CPD)

September 18, 2019

Acknowledgements

- **Duane Almeter** and **Eric Watson** for their leadership and vision on this project
- **Daniel Kikuchi** for the design of the Collect labs, the CPD install and other technical support
- **Ed Duhe, Rich Russo** and **Joshua Laing** for providing the underlying development platforms in our Dallas and Raleigh TECs
- **Anjali Shah** for her help on updating the notebook in the Analyze Part 2 lab
- **Jeff Tuck** for his help in updating his original work on the predictive modeling microservice that enriches the Stock Trader “After” application
- **Frank Ketelaars** and **Adrian Houslander** for providing input on the workshop flow and overall design
- **Ashwin Dev, Prashant Patel, Kanda Zhang, Luiz Erico Machado de Almeida, Nicholous Markey, Louis Mau** and **Sahil Shah** for CPD development team support
- **John Lucas** for final product testing and workbook publishing

For work on the original workshop, additional thanks goes to:

- **John Van Buren** for the Organize lab design
- **Kent Rubin** Analyze Part 1 lab design
- **Anjali Shah, Rui Fan** and **Ben Chard** Analyze Part 2 lab design
- **Sriram Srinivasan**, Chief Architect of the Cloud Pak for Data (CPD)and his help
- **Craig Maddux, Dean Compher, Dominic Farrar, Gary Brunell, Karen Groski, Kyle Talish, Neal Finkelstein, Patrick Pitre, Paul Betts, Ryan Kather, Sang Suh, Dale Mumper** and **David Solomon** for marketplace leadership and active participation in testing the original workshop, as well as proctoring and contributions in delivering the workshops

Special thanks to:

Beth Friday, VP, North America Technical sales - our technical executive sponsor

Hemanth Manda, Director of Platform Offerings, and **Sampada Basarkar**, Director of Development, for providing help from their teams

--

Final thanks goes to my wife Sheniqua and my son Malcolm for putting up with my taking up many evenings and more than a few weekend time-outs to complete this project.

Contents

LAB 01	INTRODUCTION AND SETUP	6
1.1	IBM JOURNEY TO CLOUD AND AI: ANALYTICS MODERNIZATION WORKSHOP	6
1.2	IBM CLOUD PRIVATE (ICP).....	6
1.3	CLOUD PAK FOR DATA (CPD).....	6
1.4	AUDIENCE FOR THIS IBM WORKSHOP.....	7
1.5	LAB WORKSHOP ENVIRONMENT.....	7
1.6	LET'S GET STARTED!	8
1.7	EXPLORE THE HOME PAGE	11
1.8	USER MANAGEMENT: PERSONA-BASED ROLES AND TEAMS	11
1.9	PROFILE SETTINGS.....	14
1.10	INSTANCES	17
1.11	LAB CONCLUSION.....	18
LAB 02	EXECUTIVE DEMO	19
2.1	GET STARTED BY IMPORTING A PROJECT	19
2.2	CONNECT TO A DATA SOURCE.....	21
2.3	STOCK TRADE OPENING BELL ANALYSIS DASHBOARD.....	23
2.4	STOCK TRADER ANALYSIS CLOSING BELL ANALYSIS	25
2.5	RUN THE STOCK TRADER "BEFORE" APPLICATION	27
2.6	RUN STOCK TRADER "AFTER" APPLICATION.....	31
2.7	LAB CONCLUSION.....	35
LAB 03	COLLECT PART 1 - CONNECT.....	36
3.1	LAB OVERVIEW	36
3.2	PERSONA REPRESENTED IN THIS LAB	36
3.3	DB2 DATA OVERVIEW – TRANSFORMING FOR ANALYTICS.....	37
3.4	MONGODB DATA OVERVIEW – VIRTUALIZING FOR ANALYTICS.....	43
3.5	LAB CONCLUSION.....	48
LAB 04	ORGANIZE	49
4.1	LAB OVERVIEW	49
4.2	PERSONA REPRESENTED IN THIS LAB	49
4.3	CREATE A BUSINESS GLOSSARY.....	50
4.4	CREATE GOVERNANCE POLICIES AND RULES.....	53
4.5	DISCOVER ASSETS	56
4.6	SHOP FOR DATA.....	61
4.7	TRANSFORM DATA.....	65
4.8	LAB CONCLUSION.....	79
LAB 05	COLLECT PART 2 - VIRTUALIZE.....	80
5.1	LAB OVERVIEW	80
5.2	PERSONA REPRESENTED IN THIS LAB	80
5.3	DATA VIRTUALIZATION DATA SOURCES.....	81
5.4	VIRTUALIZE THE MONGODB DATA WITH THE Db2 DATA	83
5.5	VIRTUALIZE THE Db2 DATA.....	86
5.6	JOIN THE TWO VIRTUALIZED TABLES.....	88
5.7	LAB CONCLUSION.....	93

LAB 06 ANALYZE PART 1 - DASHBOARDS	94
6.1 LAB OVERVIEW	94
6.2 PERSONA REPRESENTED IN THIS LAB	94
6.3 DASHBOARDS TO HELP IDENTIFY THE FOCUS AREA	94
6.4 LAB CONCLUSION.....	116
LAB 07 ANALYZE PART 2 – MODEL CREATION	117
7.1 LAB OVERVIEW.....	117
7.2 PERSONA REPRESENTED IN THIS LAB	117
7.3 TOOLSETS FOR ANALYZING DATA.....	117
7.4 WALK THROUGH A MODEL CREATION BEGINNING STEPS	118
7.5 REAL-TIME AND BATCH SCORING	129
7.6 CREATE AN EVALUATION	134
7.7 LAB CONCLUSION.....	137
LAB 08 DEPLOY AND INFUSE	138
8.1 LAB OVERVIEW.....	138
8.2 PERSONA REPRESENTED IN THIS LAB	138
8.3 DEPLOY THE MODEL	138
8.4 TEST THE DEPLOYMENT	146
8.5 THE DEPLOYMENT INFUSED IN THE APPLICATION STOCK TRADER AFTER.....	148
8.6 LAB CONCLUSION.....	151
LAB 09 WRAP-UP	152
9.1 LAB OVERVIEW.....	152
9.2 DATA SCIENTIST WRAP-UP.....	152
9.3 DATA STEWARD WRAP-UP.....	154
9.4 DATA ENGINEER WRAP-UP	155
9.5 ADMINISTRATOR WRAP-UP.....	156
9.6 WORKSHOP CONCLUSION.....	157
APPENDIX A. STOCK TRADER OPENING BELL DASHBOARD	158
BUILD THE DASHBOARD.....	158
APPENDIX B. LAB FIXES.....	171
LAB 02 – EXECUTIVE DEMO.....	171
LAB 04 – ORGANIZE.....	173

[This page left intentionally blank]

Lab 01 Introduction and Setup

1.1 IBM Journey to Cloud and AI: Analytics Modernization Workshop

This workshop provides hands-on experience with Cloud Pak for Data (CPD) that will show you how to modernize your microservices applications by enriching them with Machine Learning (ML) and Artificial Intelligence (AI).

The Journey to AI requires a strong information architecture that supports self-service capabilities and balances the needs of both the agility required by lines of business as well as the “Enterprise class” delivery required by IT. This journey can move significantly faster and with more efficiency when you use a single integrated platform like Cloud Pak for Data (CPD). It is the world’s leading platform that allows you to **Collect**, **Organize** and **Analyze** data, and then **Deploy** the results into your applications to **Infuse** them with AI.

1.2 IBM Cloud Private (ICP)

IBM Cloud Private (ICP) is a private cloud computing platform run solely for one organization. With it you can develop and manage containerized applications using the container orchestrator Kubernetes, a private image registry, a management console, and a monitoring framework.

ICP can be managed internally or by a third party, and can be hosted on premises behind your company’s firewall, or externally on a third-party provider’s cloud. Private cloud offers the benefits of a public cloud including rapid deployment and scalability, ease of use, and elasticity. Additionally, a private cloud provides greater control, increased performance, predictable costs, tighter security, and flexible management options.

IBM Cloud Private can be customized to meet the unique needs and security requirements of your organization.

1.3 Cloud Pak for Data (CPD)

Cloud Pak for Data (recently renamed from IBM Cloud Private for Data) is an integrated end-to-end analytics platform designed to help make data more accessible and trusted, as well as providing access to many analytical tools to help your organization gain insights from your data.

Cloud Pak for Data (CPD) provides the data platform that accelerates the journey up the “AI Ladder.” With it, you can quickly build, train, and deploy machine learning (ML) to create applications with Artificial intelligence (AI). CPD provides inventory and cataloging of your data sources, self-service shopping for data, and data integration and refinement capabilities. Thus, high quality, trusted data can be more easily prepared, assembled and used with this powerful integrated platform.

In this lab workshop image, CPD is installed on the foundation of ICP. However, CPD can be installed on other private cloud platforms such as OpenShift.

1.4 Audience for this IBM workshop

This IBM workshop is aimed at the line-of-business professionals who are tasked to gain new insights from all available data – regardless of its type and origin. The following personas who will be represented in the various labs will greatly benefit from this workshop:

Persona (Role)	Capabilities
Administrator	Administrators set up and maintain the CPD environment itself. The exercises in this first lab represent some typical CPD Administrator activities.
Data Engineer	Data Engineers build and optimize the systems to allow data scientists and business analysts to perform their work. The Data Engineer ensures that any data is properly received, transformed, stored, and made accessible to other users.
Data Steward	Data Stewards brings integration and governance to the data.
Business Analyst	Business Analysts deliver value by taking data, using it to answer questions, and communicating the results to help make better business decisions.
Data Scientist	Data Scientists brings expertise in statistics and the process of building ML/AI models to make predictions and answer key business questions.
Developer	Developers create and maintain the end-user applications that utilize the output from all the other personas on the CPD platform.

1.5 Lab workshop environment

We are using a small CPD cloud cluster for this workshop. This software environment was built with the following key software components:

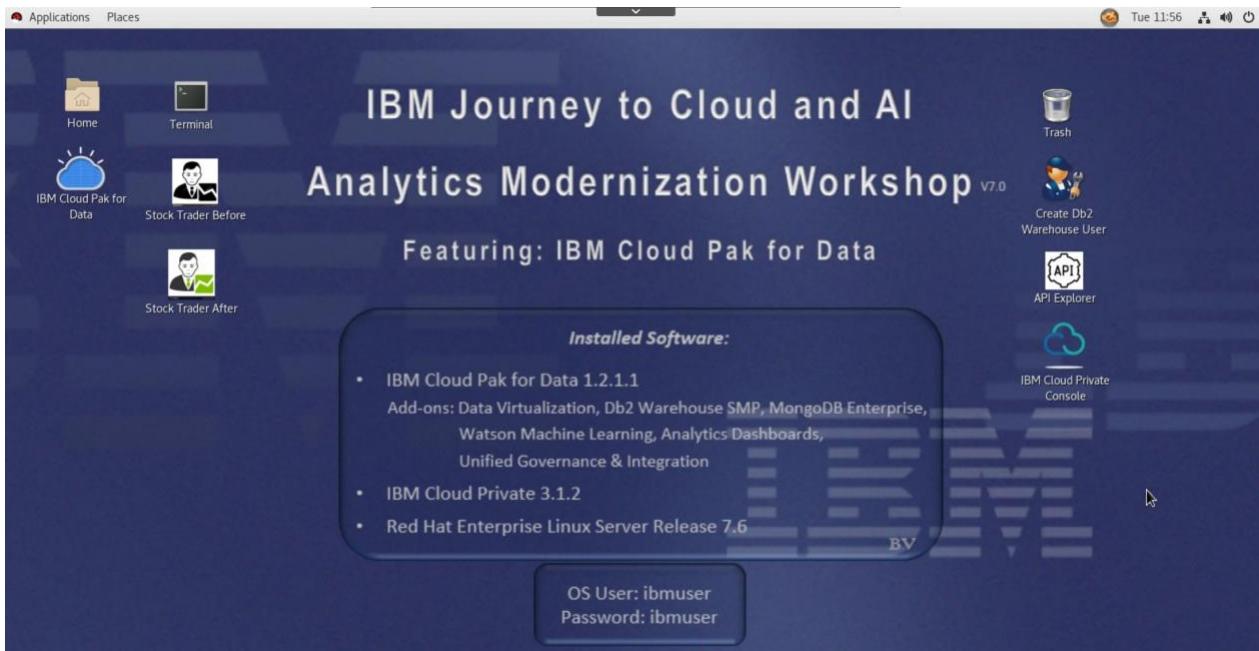
- [IBM Cloud Private 3.1.2](#) as the foundational cloud-native technology platform of Kubernetes, Docker and Helm, as well as other open-source tools.
- [Cloud Pak for Data 1.2.1.1](#) as the microservice-built, integrated data and analytics platform, installed on top of ICP, with various add-ons installed and enabled.

Note: Make sure to only click on the [node1](#) VM and then set to [Fullscreen](#)



1.6 Let's get started!

_1. On your Cloud VM, the lab desktop looks like this:

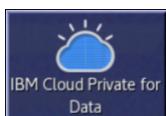


If a screensaver function locks the screen, hit [\[Enter\]](#) to get to the log in screen.
Log back in with: User [ibmuser](#), Password [ibmuser](#)

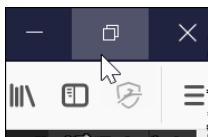
_2. Make sure to run in “Fullscreen” mode to make the most of your computer screen’s real estate.



Double-click the icon [Cloud Pak for Data](#) on your desktop to open the browser that will start the CPD web console.



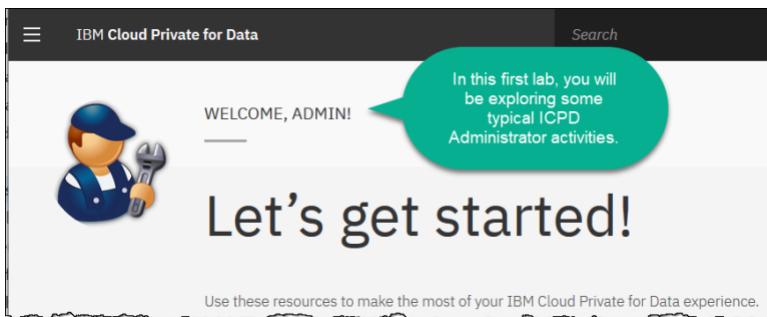
_3. After launching, maximize the browser window to make the most of your browser desktop real estate.



- 4. The CPD web console GUI displays as shown. Use **admin** and **password** for the *Username* and *Password* and click **Sign In**.

The screenshot shows the 'Sign in' page of the IBM Cloud Private for Data web interface. It has fields for 'Username' (admin) and 'Password' (password). A green checkmark is placed over the 'Sign In' button.

- 5. You should now be at the page: [Let's get started!](#)



- 6. Spend a few minutes reviewing some of the links in the page [Let's get started!](#) (**Note: you don't have to watch any of the videos.**)

- a) The **Administrator and monitor** categories provide administration tools to organize teams, define roles, provide access, and perform monitoring of the CPD cluster health.

Review the content in the links checked below - don't change, edit, or create anything.

The screenshot shows the 'Administrator and monitor' section of the 'Let's get started!' page. It includes a video player for an 'IBM Cloud Private for Data: Overview' video. To the right, there are three checked links: 'Get IBM Cloud Private for Data', 'Create users and LDAP', and 'Monitor cluster health'.

When you are done reviewing any CPD web console page in this exercise, use the back arrow key to return to the [Let's get started!](#) page.

- b) The **Collect & organize** categories provide the ability to collect data from many data sources, no matter where they reside, e.g. IBM Db2®, Oracle, Teradata, Hadoop, flat files, etc. It does not matter if data is stored in a private or public cloud, or on-premises.

It also provides a way to organize that data using embedded machine learning to assign business terms to newly discovered data, and then enforce governance policies and create a catalog of the data to enable easily prepare that data for analysis.

Review the content in the links checked below - don't change, edit, or create anything.

Administer and monitor Collect and organize Analyze

IBM Cloud Private for Data: Collect and organize

Build your enterprise data catalog and ensure that your data is mapped to a standard set of business term governance policies and rules.

- Set up data connections
- Add categories and terms
- Add policies and rules
- Explore data catalog
- Transform and integrate your data
- Create a database

Discover and add assets

- c) The **Analyze** category allows you to build predictive and prescriptive models using open source programming languages (typically for the data scientist) or using graphical user interface tooling (typically for the business analyst). With the click of a button, deploy these models into production and publish an Open REST API that can be consumed by any application running on IBM Cloud Private or any other platform. Build dashboards and interact with your data to gain insight and business understanding.

Review the content in the links checked below - don't change, edit, or create anything.

Administer and monitor Collect and organize Analyze

Unlock insights from your data with dashboards, notebooks, RStudio, and a machine learning model builder.

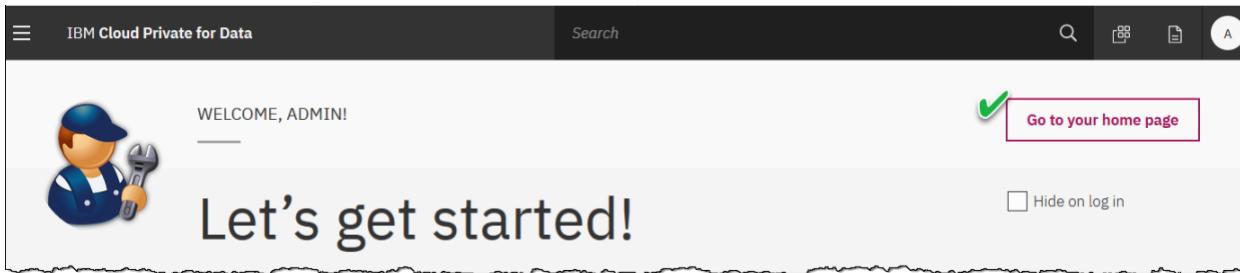
- Create an analytics project
- Create an analytics dashboard



In this workshop, we will demonstrate the **Collect**, **Organize** and **Analyze** capabilities to create a machine learning model that can be deployed for consumption by a microservice application.

1.7 Explore the home page

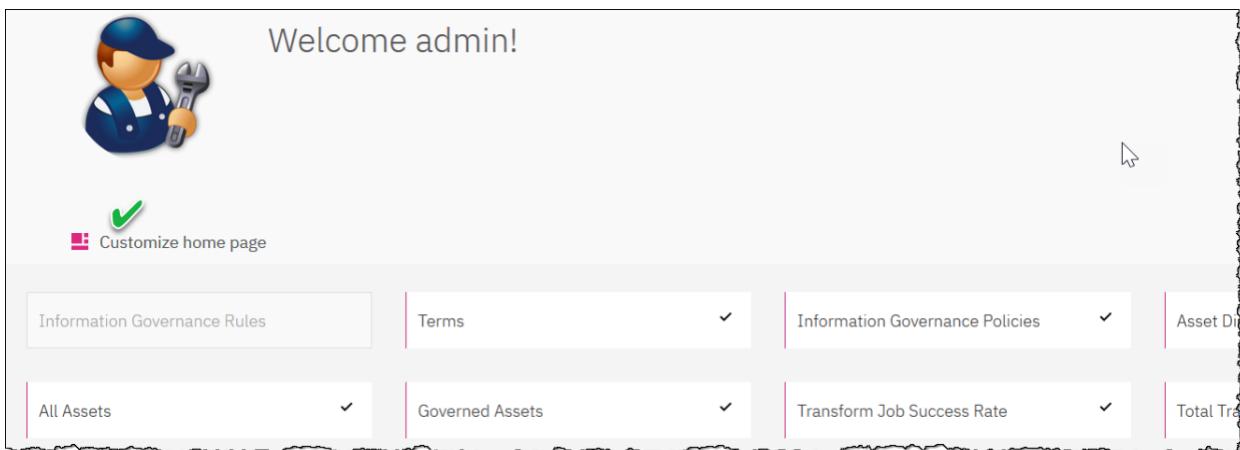
- __7. From the [Let's get started](#) page, click on [Go to your home page](#).



- __8. Click on [Customize home page](#)

Notice the home page shows the inventory of the integrated data platform for Information Governance Rules, Terms and Policies, Asset Distribution, All Assets, Governed Assets, Transform Job Success Rate, Total Transform Job Executions. The intent of this page is to provide individual customizations on daily tasks for each CPD user.

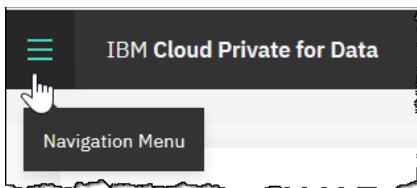
Scroll through the page to see inventory and asset types. Since we are starting fresh, there is not a lot to see now, but we can visit the home page later to review the list of assets after we have created new ones.



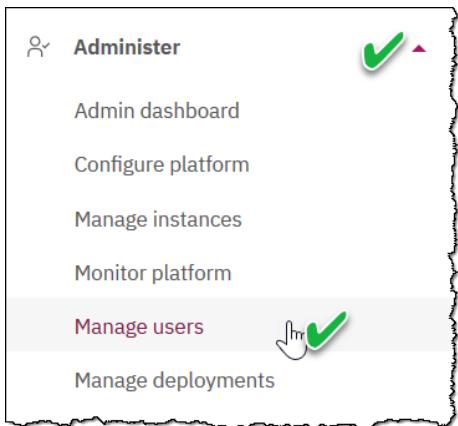
1.8 User management: Persona-based roles and teams

This section explores user authorizations for the various stages of the data analytics pipeline.

- __9. Click on the hamburger icon on the top left part of the screen to access the [Navigation Menu](#):



- __10. Click on the [Administer](#) twistie to display the drop-down menu, and then click on [Manage users](#)



- __11. Click on the [Roles](#) section to review the various personas that can be represented by any given user. A user can be granted more than one role if needed.

Role	Description
Administrator	Administrator role
Business Analyst	Business analyst role
Data Engineer	Data engineer role
Data Scientist	Data scientist role
Data Steward	Data steward role
Developer	Developer role



Admin

These existing roles can be customized, or new ones created, to suit your company's needs. CPD is very much "persona driven" in that each user can play their particular part in your organization's journey to AI. Each user (acting as one or more personas) can hand off and/or share their work with other users/personas, for a totally collaborative environment.

12. Click back to the [Users](#) section and then click on [Connect to an LDAP Server](#).

The screenshot shows the 'Manage users' page. At the top, there are two tabs: 'Users' (which is selected) and 'Roles'. Below the tabs, there is a green checkmark icon and the word 'Users'. On the right side of the header, there is another green checkmark icon and the text 'Connect to an LDAP server'. Below the header, there is a search bar labeled 'Filter Users' with a magnifying glass icon. A table below the search bar lists six users with columns: Name, Status, Username, Date added, User id, and Roles. Each user row has a small green checkmark icon at the end.

Name	Status	Username	Date added	User id	Roles
Business Analyst	Approved	businessanalyst	06/27/2019, 3:21 PM	1002	Business Analyst
Data Engineer	Approved	dataengineer	06/27/2019, 3:20 PM	1001	Data Engineer
Data Scientist	Approved	datascientist	06/27/2019, 3:22 PM	1003	Data Scientist
Data Steward	Approved	datasteward	06/27/2019, 3:23 PM	1004	Data Steward
Developer	Approved	developer	06/27/2019, 3:23 PM	1005	Developer
admin	Approved	admin	--	999	Administrator + 5 more

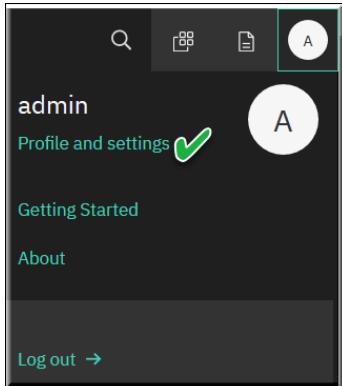
Notice that you can integrate with your LDAP server by creating a connection to it using “with search” or “without search” approaches.

The screenshot shows a configuration form for connecting to an LDAP server. The title is 'Connect to your LDAP server'. There are two radio buttons: one selected (filled purple) labeled 'with search' and one unselected (outline purple) labeled 'without search'. Below the radio buttons is a section titled 'LDAP URL' with the sub-instruction 'The host name with the ldap:// or ldaps:// protocol'. Below that is a section titled 'LDAP port' with the sub-instruction 'The port for the LDAP connection (typically 389 for ldap and 636 for ldaps)'. At the bottom is a section titled 'Domain search user' with the sub-instruction 'The LDAP user that performs user lookups'.

1.9 Profile settings

- 13. Click the top right circle of your screen. Currently, it has the letter “A” on it, which is the first letter of “admin.” If you were to log in with a user name like “ibmuser” this circle would say “I”.

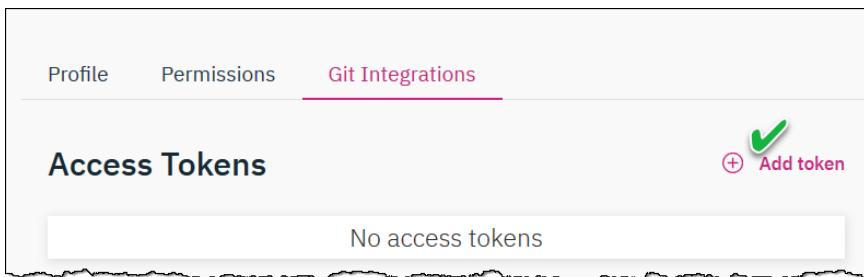
Choose [Profile and settings](#)



- 14. Then review the things you can change in your [Profile](#), then review [Permissions](#), then go to [Git Integrations](#) (Note: in Permissions you have all permissions because you are the Admin user. A non admin user would NOT have all these privileges.)



- 15. In [Git Integrations](#) ⇒ [Add token](#).



- __16. Notice that you can manage your repository with [GitHub](#), [GitHub Enterprise](#), [BitBucket](#) and [BitBucket Server](#).

Add token

Visit [GitHub personal access tokens](#), select repository scope and generate a token.

Platform*

GitHub GitHub Enterprise BitBucket BitBucket Server

Access Token*

Paste generated personal access token here

This capability allows you to integrate CPD projects with your current CICD (Continuous Integration and Continuous Delivery) pipeline to automate delivery of the artifacts you create in the CPD platform. You can use capabilities from the underlying IBM Cloud Private platform to build cloud native microservice applications which are tied to the ML / AI model development and delivery pipeline.

- __17. Click the [Add-on](#) icon (four little squares over one bigger square) on the top right corner of your screen.



- __18. This will bring up all available add-on products for CPD.

Choose [Data sources](#) from the menu (or scroll down to find this section) and notice that [Db2 Warehouse SMP](#), [Data Virtualization](#) and [MongoDB Enterprise](#) have been “enabled,” which means they will be available to be used in this workshop.

Add-ons

All categories

- AI
- Analytics
- Dashboards
- Data governance
- Data sources**
- Developer tools
- Industry accelerators
- Storage

Data sources

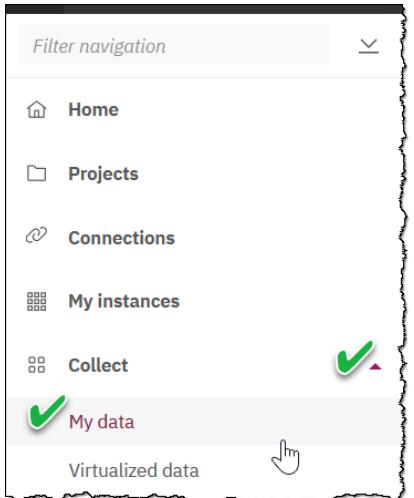
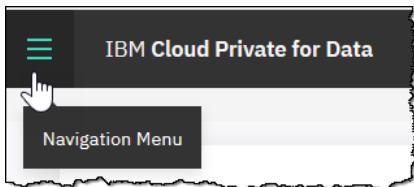
CockroachDB Partner Premium CockroachDB: Ultra-resilient SQL for global business	Db2 Event Store IBM Premium In-memory data store capable of extremely high speed ingest and deep, real-time analytics.	Db2 Advanced Enterprise Server Edition IBM Premium Relational database that delivers advanced data management and analytics capabilities for transactional and warehousing workloads.	Db2 Warehouse MPP IBM Data warehouse MPP designed for high-performance, in-database analytics.
Db2 Warehouse SMP IBM Enabled Data warehouse designed for high-performance, in-database analytics.	IBM Db2 for z/OS IBM Create databases in Db2 for z/OS and work directly with the data from IBM Cloud Private for Data	Data Virtualization IBM Enabled Query many data sources as one.	MongoDB Enterprise Partner Enabled Premium Scalable, open source NoSQL database.



Admin

There are currently many other add-ons available for your organization's use at the click of a button, and IBM continues to add more with each release of Cloud Pak for Data.

- _19. To review the Db2 Warehouse that has been installed, click on the [Navigation Menu](#) hamburger icon \Rightarrow [Collect](#) \Rightarrow [My data](#)



- _20. Choose the [Databases](#) section to see [Db2WarehouseSMP](#).

Click the ellipses (the three stacked dots) then [Open](#) to launch the *Db2 Data Server Manager* console.

Database Name	Type	Status	Created On
MongoDB-Activity1	MongoDB Enterprise	Available	Created on 22 May 2019
MongoDB-Activity2	MongoDB Enterprise	Available	Created on 22 May 2019
Db2Warehouse	Db2 Warehouse SMP	Available	Created on 13 May 2019

21. From here, many options are available to you for using this database through the hamburger menu icon. Choose it to do a quick review the capabilities of this Db2 Warehouse GUI.



We will be using this database add-on in later exercises as a different persona, so there is no need to explore further now.

 Admin	<p>Applications and databases available through add-ons are native to, and integrated on, the CPD platform.</p> <p>Besides using its own products, IBM partners with MongoDB, portworx, Datameer, Lightbend, Senzing Prolifics and others, to build a rich ecosystem around the CPD platform.</p> <p>Details can be found at https://www.ibm.com/products/cloud-private-for-data/partners</p>
---	--

1.10 Instances

22. Click on the hamburger icon to access the **Navigation Menu** ⇒ **My Instances**



- _23. Click tab **Provisioned Instances** and then click the twistie to sort the instances of Data Virtualization, Db2Warhouse, and MongoDB databases that were provisioned for this workshop:

My Instances

Name	Type	Provisioned by
Data Virtualization	dv	user999
Db2WarehouseSMP	db2whsmp	user999
MongoDB-Activity1	mongodb	user999
MongoDB-Activity2	mongodb	user999

1.11 Lab conclusion

Cloud Pak for Data is most useful in the following business use-case scenarios:

1. **Manage All Your Data:** Use discovery to automate the process of assigning business terms to technical assets. Use enterprise catalog to secure, govern and control access to your data regardless where it resides. Use a combination of virtualization and transformation to prepare your data for analysis.
2. **Build Your Ladder to AI:** Build a strong information architecture to help you realize the value of leveraging machine learning and AI.
3. **Modernize Your Data & Analytics Workloads:** Modernize your data platform and provide data scientists and application developers the ability to quickly add AI to your applications. Since ICP is the foundation of CPD, you can use ICP to build cloud-native, microservice-based applications and use CPD to enrich them with machine learning and artificial intelligence.
4. **Compliance Readiness** – Our strong governance capabilities allow you to provide regulatory compliance.



Make Cloud Pak for Data your platform for data and analytics. Why? Because IBM understands data and provides an integrated, end-to-end data platform that enables enterprises to:

- Collect relevant data and make it simpler and more accessible
- Use federation, virtualization and/or transformation to combine and refine data sets
- Organize data so it can be trusted
- Analyze insights on demand
- Infuse machine learning in your applications

All of the above will be demonstrated in the following workshop labs.

** End of Lab 01 – Introduction and Setup

Lab 02 Executive Demo

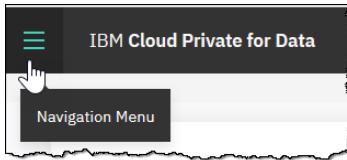
The key purpose of this lab is to see what an application called Stock Trader could look like before and after it has been infused with a machine learning (ML) model created from the CPD platform.

Before you begin to use CPD to create something, this lab shows you what can be accomplished with it.

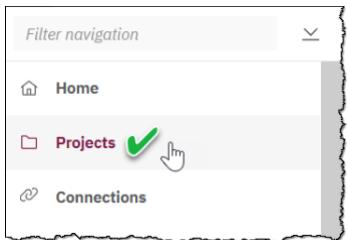
2.1 Get started by importing a project

To get started, we will work with a “project” that has been created for this workshop which was exported to a zip file. As you will see in more detail later, a project is a useful collaborative CPD construct that allows a team to work together on many assets with appropriate access controls.

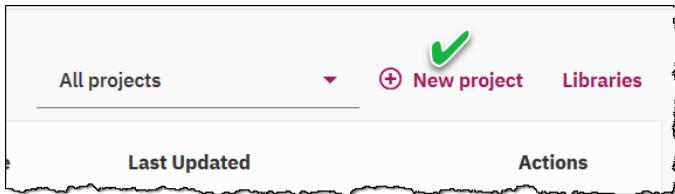
- _1. In the CPD console, click on the hamburger icon to access the [Navigation Menu](#):



Then choose [Projects](#)



- _2. Click [New project](#).



- _3. Give the project the name [TradingCustomerChurn](#).

Note: case is important for this exercise, please enter the project name exactly as shown and then click [OK](#).

Create a new project

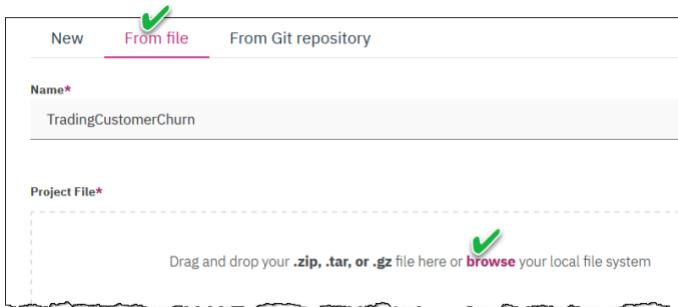
Analytics project Data Transform project

Project name*
TradingCustomerChurn

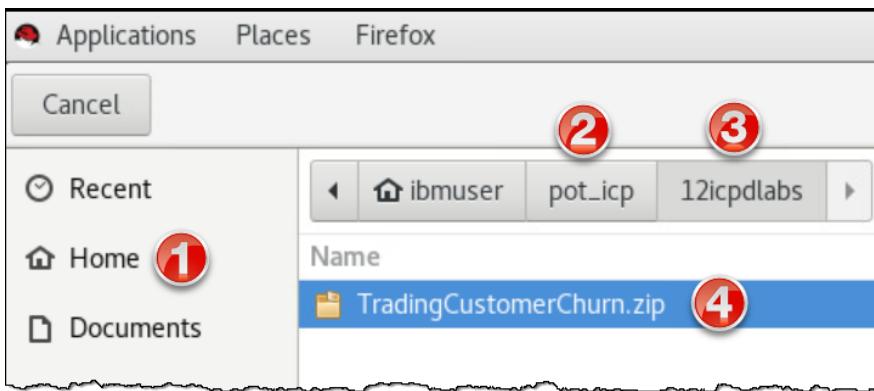
Note: please type this exactly as shown

Cancel OK

- 4. Click the middle tab **From file** and then click the link **browse**.



- 5. On the left pane, double-click **Home** (which is the home for *ibmuser*) then choose directory **pot_icp** then directory **12icpdlabs**. Finally, double click file **TradingCustomerChurn.zip**



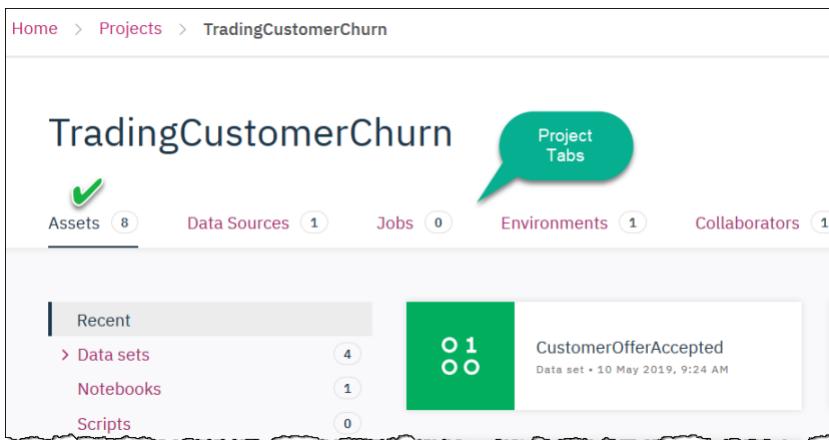
- 6. Click **Create** (towards the bottom right) to create the new project by importing the zip file.

Note: If there is a problem with importing the zip file. Delete and retry the import.



The project creation process launches multiple resources in the CPD cluster, which may take a minute or two – be patient while it completes.

- 7. When it is finished, you will be presented with a project navigation screen that is at first positioned at the **Assets** project tab. Notice there are other tabs like **Data Sources**, **Jobs**, etc.



2.2 Connect to a Data Source

- 8. Next, click the tab called [Data Sources](#) and then [Db2Warehouse](#).

The screenshot shows the 'Data Sources' tab selected in a navigation bar with counts (1, 0, 1, 0). Below is a search bar and a table titled 'Data Sources'. A single row is listed: 'Name' (Db2Warehouse) with a green checkmark icon next to it.



You might recall that in this lab environment we installed Db2 Warehouse as an add-on feature, indicating that it has tight integration with CPD and is cloud-native. This is the case for all add-ons that you choose to deploy in CPD.

- 9. Scroll down and specify *Username*: [icpd](#) and *Password*: [icpd](#)
Click [Test Connection](#) to make sure your credentials work.

The form fields are as follows:

- Data source type: Db2
- JDBC URL: jdbc:db2://db2whsmp-1557795192.db2whsmp.svc.cluster.local:50000/BLUDB
- Username: icpd (with a green checkmark)
- Password: [icpd](#) (with a green checkmark)
- Shared:
- Test Connection: A button with a green checkmark icon and a pink border.

__10. You should get [Test connection succeeded](#).

If not, retype the username/password again until you get this message... and make sure your case is correct!

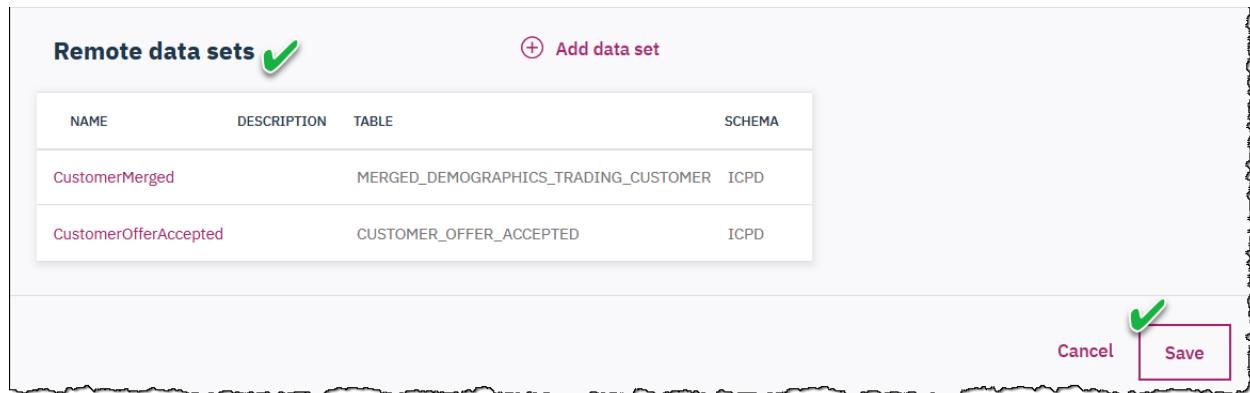
 Test connection succeeded.



If the [Test connection](#) fails with a hostname issue, see Appendix B – Lab fixes – Lab 02 “Test connection fails”

__11. Scroll down to review the two [Remote data sets](#) already in the project. In this case, these particular “data sets” are actually tables in the database

Before you exit this screen, save the Username/Password changes you did earlier in this screen by clicking [Save](#).



NAME	DESCRIPTION	TABLE	SCHEMA
CustomerMerged		MERGED_DEMOGRAPHICS_TRADING_CUSTOMER	ICPD
CustomerOfferAccepted		CUSTOMER_OFFER_ACCEPTED	ICPD

Cancel  Save



If you have trouble rendering the dashboard later in this lab, that is because you did not save the credentials to this data source and did not test the credentials properly.

2.3 Stock Trade Opening Bell Analysis dashboard

This is the business use case scenario: Boatswain Trading is aware that revenues are declining. They have used CPD to create various assets to help them with this problem.

To begin, we will discover the trend of that revenue decline by reviewing some simple visualizations. In this scenario, the business analyst has requested aggregated stock transaction data. It contains data such as the historical total number of individual customer visits to their website as well as the number of trades per customer for the past year.

- 12. Let's take a look at an example dashboard that the business analyst has already created. (Note: in a later lab, you will build a dashboard from scratch.)

In the project, click the [Assets](#) tab, then the [Analytics dashboards](#) option:

The screenshot shows the CPD interface with the 'TradingCustomerChurn' project selected. The 'Assets' tab is active, and the 'Analytics dashboards' option under the 'Analytics dashboards' section is highlighted with a green checkmark. A list of three pre-built dashboards is displayed: '03-Stock-Trader-Closing-Bell', '02-Stock-Trader-Demographic-Discovery', and '01-Stock-Trade-Opening-Bell'. A tooltip 'Click to sort' points to the 'Name' sorting arrow.

- 13. Notice we have three pre-built dashboards. Click on the up/down arrows next to [Name](#) to sort them.

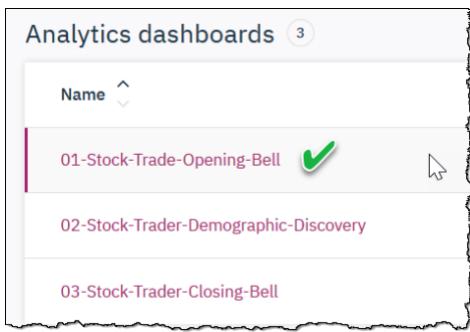
The screenshot shows a list of three analytics dashboards: '01-Stock-Trade-Opening-Bell', '02-Stock-Trader-Demographic-Discovery', and '03-Stock-Trader-Closing-Bell'. A tooltip 'Click to sort' points to the 'Name' sorting arrow.

- 14. The first dashboard ([01...](#)) shows a visualization of the 'Stock Trade Before' analysis.

The second dashboard ([02...](#)) shows the results of the analysis of the customer demographics data that was collected from various sources (such as Db2 Warehouse and MongoDB) after curation of the data. This information is used to evaluate which measures (or factors, or variables) have the greatest impact on the customer churn.

The third dashboard ([03...](#)) shows the visualization of the Stock Trader application after implementation of the ML model and the impact that it had on the business.

- __15. To view the first dashboard, click on it: [01-Stock-Trade-Opening-Bell](#).



- __16. This dashboard shows the consolidated report of the business in shares sold and traders. You can see that the number of shares sold per month is relatively flat, and worse, daily traders are declining.



Hint: to adjust the zoom on your browser to see more or less of any dashboard, try using [Ctrl][Mouse-scroll-wheel]

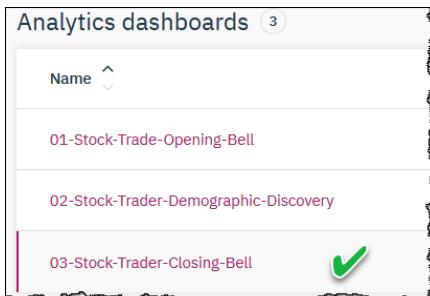
	In CPD, the data for this report is stored at a central place and is governed and managed with team collaboration between business analysts, data engineers, data scientists and data stewards.
--	---

Note: See *Appendix A* for details on building this dashboard from scratch.

2.4 Stock Trader Analysis Closing Bell analysis

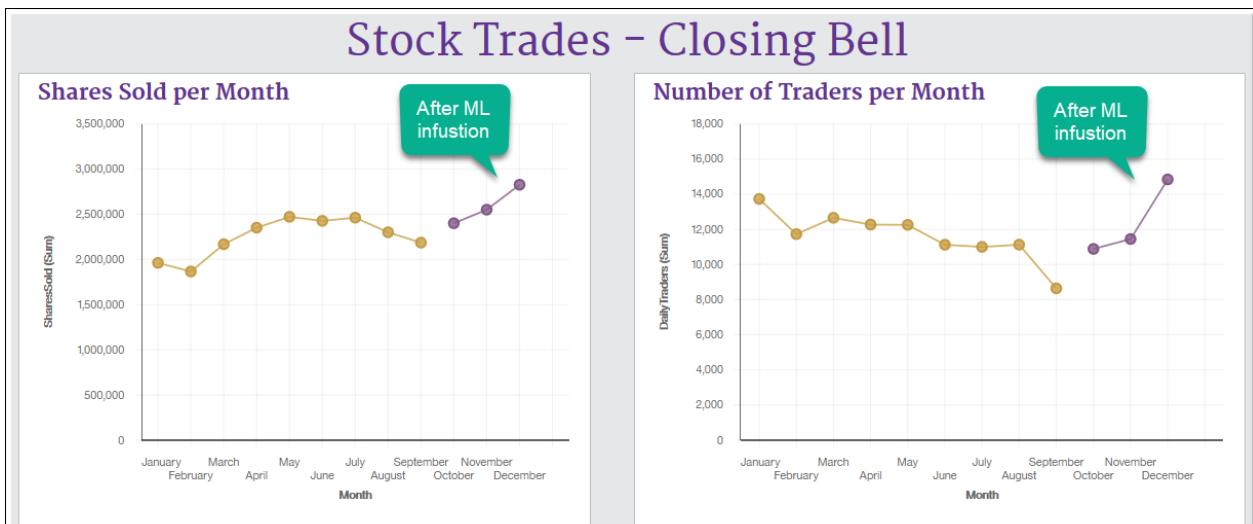
After completing the review of how the business was looking in the beginning, let's review the same information after Boatswain Trading infused a Machine Learning model into their Stock Trader application to improve its business.

- _17. Click on the browser back arrow key to return to your [Analytics dashboards](#) options. 
- _18. Now choose dashboard: [03-Stock-Trader-Closing-Bell](#)



	If the dashboard does not render, see Appendix B – Lab Fixes - Lab 02 – “Dashboard doesn’t render.”
---	---

- _19. Notice the visualization trend of [Shares Sold per Month](#) and [Number of Traders per Month](#) match with the Opening Bell dashboard we just reviewed... at least, up until September.



The data from October on represents the results of Boatswain Trading's microservice application that was infused with a machine learning model. This model was built and scored in CPD using machine learning algorithms that allowed the application to offer personalized promotions based on their history and risk of churn (leaving Boatswain Trading for another company.)

The results show that the following months after implementation of this new model, they had a large increase in both the number of shares sold per month, as well as the number of stock traders per month.

Boatswain's Trading business is now increasing rather than decreasing – this means the initiative was a resounding success!

- 20. To be fair, this success did come with a cost: the free trades and consultations.

Scroll down in the dashboard to see the lower set of visualizations that give us insight into what that cost turned out to be. Note the total number of retention offers accepted and the total cost of those offers, which are summarized and broken down by the offer accepted “type.”

Given the increased number of “shares sold per month” as well as “traders per month” Boatswain executives can calculate that the \$15,600 USD “opportunity cost” of those free trades and free consultations is a very good trade-off for increasing their revenue overall at a much higher rate than the cost. Besides that, free trades do not really cost Boatswain Trading in actual dollars, but it can be written off as opportunity cost on the books.

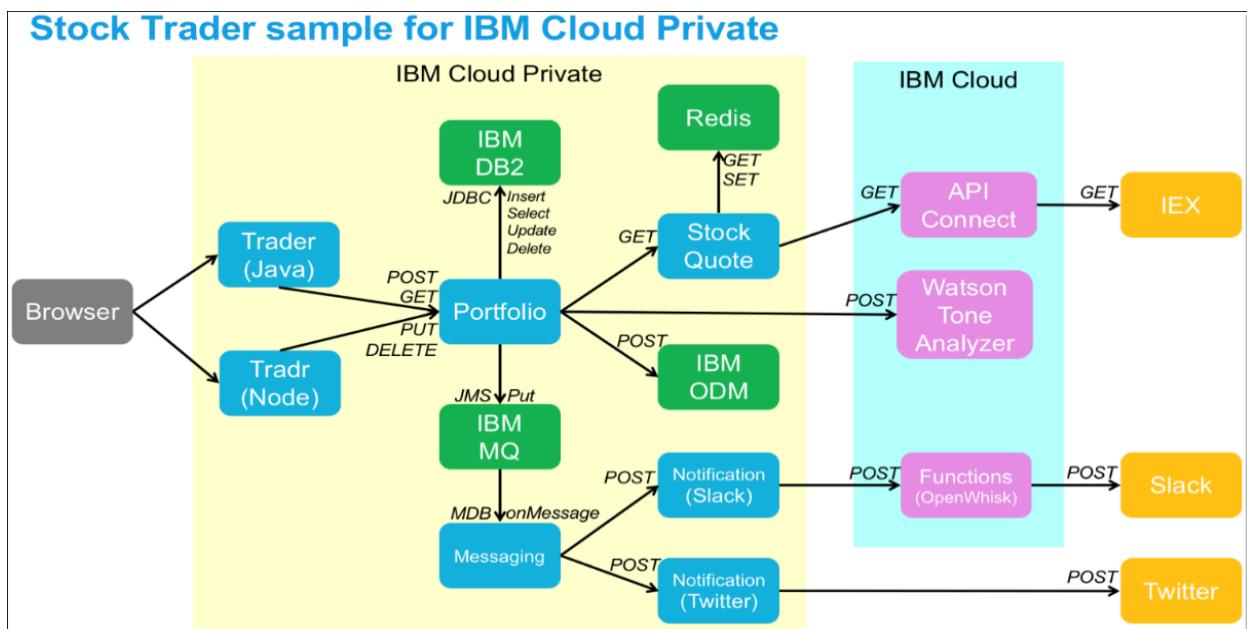


2.5 Run the Stock Trader “Before” application

Since 2015, software development and distribution methods and deployment have been changing rapidly. The Stock Trader application that we show here was built on a microservices-based architecture that allows loosely coupled services to work together and allows components to be upgraded incrementally, as opposed to traditionally with monolithic application development.

The sample Stock Trader application architecture is displayed below showing interaction between various microservices. The infrastructure services Db2, MQ, Redis and ODM are shown in green boxes, whereas microservices are shown in blue. IBM Cloud APIs are shown in pink and messaging endpoints are in orange.

The main advantage of microservice-based application is the ability to modify, deploy, update and scale services dynamically and individually. In a modern microservice mesh architecture, it is possible to run multiple versions of the same service in production and control the traffic dynamically based on rules that can be defined outside of the service itself, eliminating the need to need to modify any of the microservice code.



The Stock Trader application shown here is simply a demo application that does not address all features and functions of a commercial application, but does give insight into the concept of application modernization. The IBM Cloud Private platform provides tools to help you migrate your monolithic applications to a microservices-based architecture, free of cost.

If your journey on this path has not yet begun, share this application with your development teams. It will provide your organization with the framework you need to start building your own microservice-based applications. Find it at: <https://github.com/IBMStockTrader>

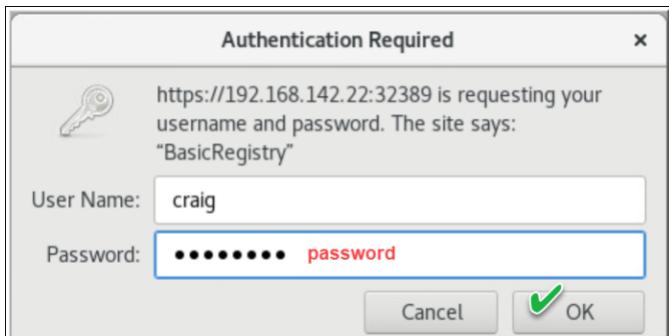
	<p>Another advantage of a microservices-based application is that it allows polyglot services to be written in the language of choice by the developers. This allows services and applications to work together easily.</p> <p>Developers are no longer forced to change their preferred language or learn a new language due to monolithic legacy applications.</p>
---	--

Let's launch this application.

- _21. On your workshop desktop, find and double-click on the icon: **Stock Trader Before**



- _22. A login popup text box displays – enter User Name: **craig** and Password: **password**.



NOTE: The first time the Stock Trader application is launched it may take a little longer to start than subsequent launches.

- _23. Select **Create a new portfolio** and click **Submit**.

- _24. Type **TechStocks** as name of the portfolio and click **Submit**.

_25. Notice Loyalty Level is Basic.

Select [Update selected portfolio](#) and click [Submit](#).

The screenshot shows the 'Summary' page of the IBM TRADER application. At the top, there are three tabs: 'Home', 'Summary' (which is active), and 'Add Portfolio'. Below the tabs, the word 'Summary' is displayed in a large font. A welcome message 'Welcome to IBM Trader powered by ICP for Data' follows. There is a list of actions with radio buttons: 'Create a new portfolio', 'Retrieve selected portfolio', 'Update selected portfolio (add stock)' (which is selected and highlighted with a green checkmark), and 'Delete selected portfolio'. Below this is a table with three columns: 'Owner', 'Total', and 'Loyalty Level'. A single row shows 'TechStocks' as the owner, '\$0' as the total, and 'Basic' as the loyalty level. A green arrow points to the 'Basic' text. At the bottom, there are two buttons: 'Submit' (with a green checkmark) and 'Change User'.

_26. Type [IBM](#) and [1000](#) and click [Submit](#) to buy 1000 stocks of IBM.

The screenshot shows the 'Add Stock' page of the IBM TRADER application. At the top, there are three tabs: 'Home', 'Summary' (active), and 'Add Portfolio'. Below the tabs, the word 'Add Stock' is displayed in a large font. A welcome message 'Welcome to IBM Trader powered by ICP for Data' follows. There are three input fields: 'Owner' (empty), 'Stock Symbol' containing 'IBM' (highlighted with a green oval), and 'Number of Shares' containing '1000' (highlighted with a blue rectangle). At the bottom, there is a 'Submit' button with a green checkmark.



If internet access is available – the stock quotes are taken from quandl.com using the IBM API Connect through IBM Cloud.

The cached values from Quandl are stored in a Redis database.

- _27. Notice that the loyalty level changed to Gold.

The screenshot shows the 'Summary' page of the IBM Stock Trader application. At the top, there are three tabs: 'Home', 'Summary' (which is selected), and 'Add Portfolio'. Below the tabs, the word 'Summary' is displayed in a large font. A welcome message 'Welcome to IBM Trader powered by ICP for Data' follows. There is a list of options with radio buttons: 'Create a new portfolio' (unchecked), 'Retrieve selected portfolio' (checked), 'Update selected portfolio (add stock)' (unchecked), and 'Delete selected portfolio' (unchecked). To the right of this list is an advertisement for 'IBM Cloud for Data' with the text 'Cloud', 'Big Data', 'Machine Learning', 'AI', and 'No assets required'. Below the options is a table with three columns: 'Owner', 'Total', and 'Loyalty Level'. The first row shows 'TechStocks' as the owner, '\$129,690' as the total, and 'Gold' as the loyalty level. A teal arrow points to the 'Gold' entry.

Owner	Total	Loyalty Level
TechStocks	\$129,690	Gold

- _28. Final note: as part of this application, there are "Ingress" rules defined such that the application's **notification-service** can trigger a notification to either **Slack** or **Twitter**.

In this case, the **notification-service** sent our stock purchase transaction to Twitter (if the internet is available to this service) - you can check it out by going here: <https://twitter.com/ibmstocktrader>.

The screenshot shows a tweet from the account @IBMSocketTrader. The tweet reads: 'On Monday, May 13, 2019 at 12:58 PM UTC, DemoRock changed status from PLATINUM to GOLD. #IBMSocketTrader'. Below the tweet are four small icons: a speech bubble, a retweet symbol, a heart, and an envelope.

- _29. Close the **Stock Trader Before** browser window to finish the review of this application.

	<p>The Boatswain Trading goal is to enhance their Stock Trader application with analytics modernization to increase revenue and profits. They did so and named their new application: Stock Trader After</p> <p>This modern Stock Trader application was built on the modern IBM Cloud Private platform uses agile technologies with a modern CICD pipeline to implement changes rapidly.</p>
--	---

2.6 Run Stock Trader “After” application

Companies that are able to keep pace with the technological breakthroughs will have a competitive advantage over companies that do not. (Think Netflix vs. Blockbuster)

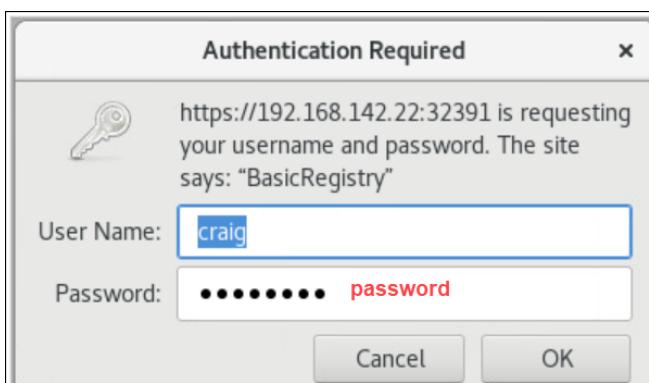
Modernization of your application is the essence of ICP, and CPD goes one step further – it allows you to use your organization’s data to modernize its analytics capabilities to infuse ML / AI into those applications.

- __30. So, let’s look at the finished product of the newly modernized, ML-infused application [Stock Trader After](#).

Find and double-click desktop icon: [Stock Trader After](#).



- __31. Log in using Username: [craig](#) and Password: [password](#) ⇨ OK



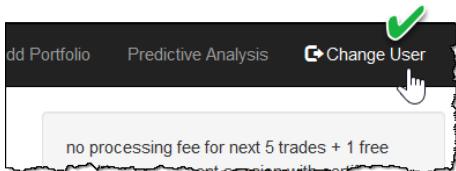
- __32. Notice the offer given for this user. (Note: Your offer may vary)

Owner	Total	Loyalty Level
TechStocks	\$129,690	Gold

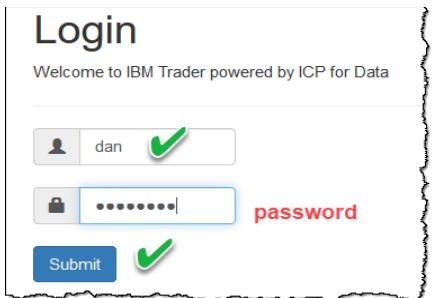
Boatswain Trading used a predictive model that looks for the score of the user [craig](#) and returns a number of parameters. Our new predictive analysis microservice uses a single parameter and transforms it into an offer to the customer based on the "separation risk" that the model predicts.

How this works will become more evident as you work through this workshop.

__33. Click [Change User](#).

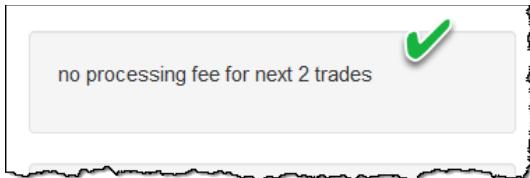


__34. Specify credentials: [dan](#) and [password](#).



__35. Notice the personalized award offer.

You can try other users like jeff, anjali, vikram (all have the same password)



__36. Let's dive deeper behind the scenes of machine learning model prediction.

Click [Predictive Analysis](#).



__37. This screen is built to directly interact with the machine learning model you will create in the workshop.

You can change the variables to see how those specific sets of variables impact predictions, but in your first test just use the defaults.

Scroll down and click [Submit](#) to assess the separation risk based these [default](#) parameters.

Customer Churn Predictor

Welcome to IBM Trainer powered by ICP for Data

Age:

Gender: Male Female

Marital Status: Married Single

Number of Children:

Home Owner: Yes No

Estimated Income:

Net Realized Gains YTD:

Net Realized Losses YTD:

Smallest Single Transaction:

Largest Single Transaction:

Total Dollar Value Traded:

Total Units Traded:

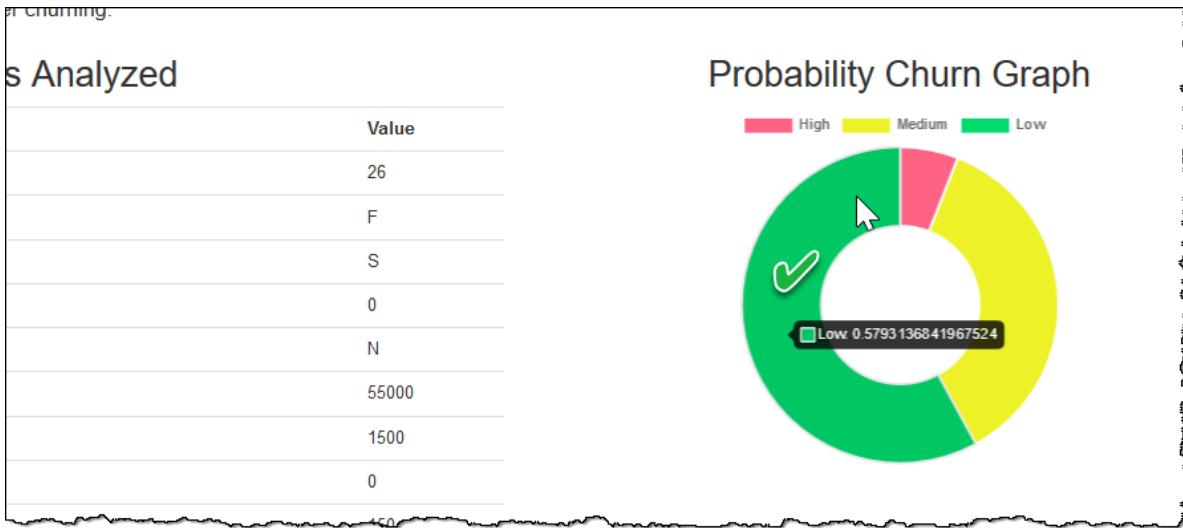
Days Since Last Login:

Days Since Last Trade:

Percentage Change Calculation:

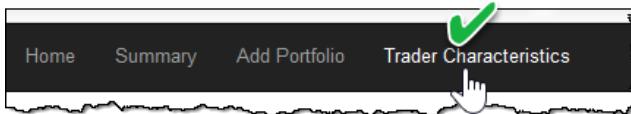
 Submit

- __38. This is the result using the default values of separation risk predictor. In this case, there is a *Low* to moderate risk of this customer leaving. This customer has a profitable portfolio.



- __39. Let's change some parameters.

Click [Trader Characteristics](#).



- __40. Change [Net Realized Gains YTD](#) from [1500](#) to [0](#) and change [Net Realized Losses YTD](#) to [1500](#) from [0](#). This represents a portfolio that is not profitable.

Scroll to the bottom and click [Submit](#).

The form contains the following fields:

- Estimated Income: 55000
- Net Realized Gains YTD: 0 (with a green checkmark)
- Net Realized Losses YTD: 1500 (with a green checkmark)
- Smallest Single Transaction: 150

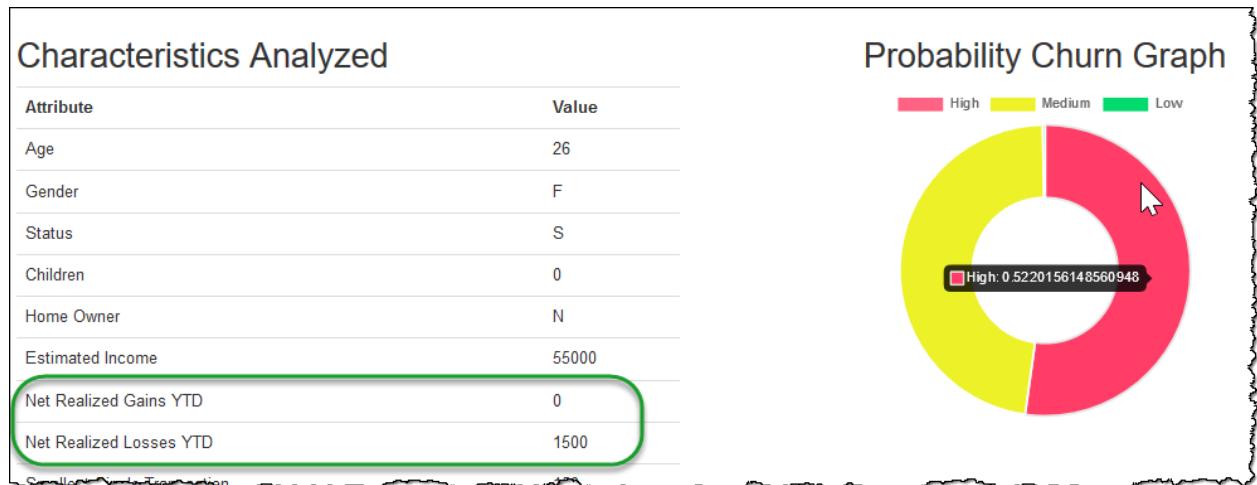
Two callout boxes provide feedback on the changes:

- A green callout box next to the "Net Realized Gains YTD" field says "change from 1,500 to 0".
- A green callout box next to the "Net Realized Losses YTD" field says "change from 0 to 1,500".

At the bottom left is a blue "Submit" button.

- 41. Notice the probability of separation risk of this customer. The risk has changes from **Low** to **Medium** given the fact that portfolio of this customer is no longer profitable.

In fact, the details show that this customer is on the cusp of being **High** probability of churn.



2.7 Lab conclusion

Regardless of your problem classification domain, CPD provides the tools needed to access, clean, shape and govern your data in preparation for ML model development, release and continuous improvement framework.

From automatic ML model generation (no coding) to comprehensive tool sets to develop complex models (with coding), CPD provides the tools necessary to produce your specific ML model classification.

Once ML models are deployed, they are easily consumed by applications as RESTful-compliant services. The deployed models are easily scalable to meet usage demands as they are packaged as independent scalable docker containers.

As probabilities are consumed from one or more models, application logic can be applied to allow applications to make intelligent business decisions based on a set of predefined business rules and workflows.

You will next dive deeper into the **Collect, Organize, Analyze** and **Deploy** capabilities of CPD as an integrated data platform.

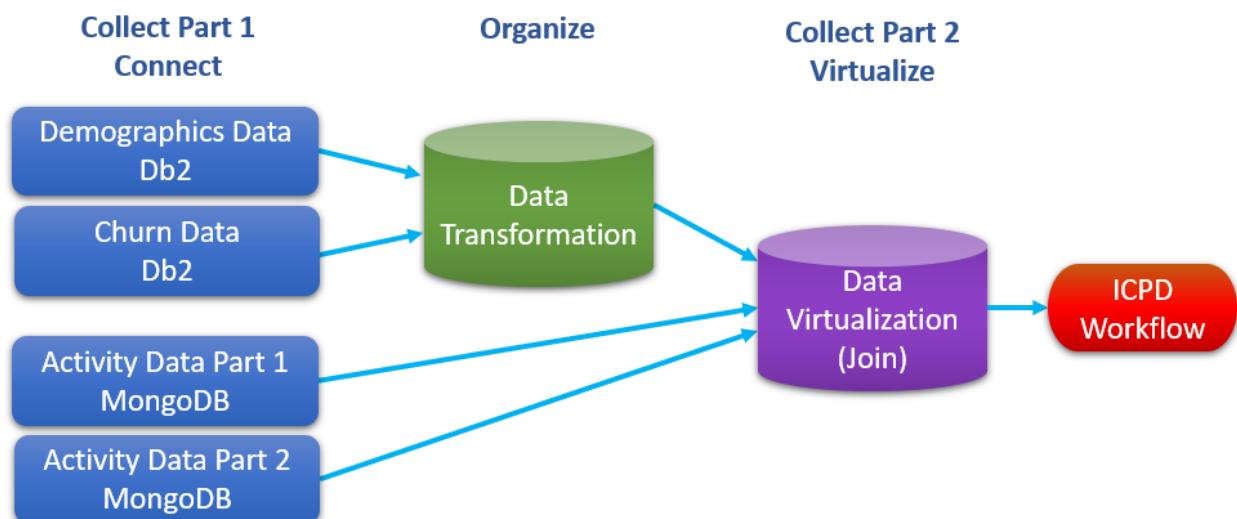
**** End of Lab 02: Executive Demo**

Lab 03 Collect Part 1 - Connect

3.1 Lab overview

The [Collect](#) capability of CPD means accessing your organization's data regardless of where it resides, whether that be a in-cluster data source (e.g. the Db2 Warehouse), a Federated data source in a remote location, or a native connection to a remote data source. You can even use the Db2 Event Store add-on feature to provide streaming access that is suited for Internet of Things processing. Additionally, Data Virtualization and Data Transformation are available real-time to streamline the access, performance, and formatting of the data for use in later steps of the CPD analytics workflow.

In this lab you will explore creating [Connections](#) and data sets for the [Collect](#) process. In later labs you will explore [Data Transformation](#) and [Virtualization](#).



3.2 Persona represented in this lab

The [Data Engineer](#) persona is the likely role to perform the various [Collect](#) tasks shown in this lab.

Persona (Role)	Capabilities
 Data Engineer	Data Engineers build and optimize the systems to allow data scientists and business analysts to perform their work. The Data Engineer ensures that any data is properly received, transformed, stored, and made accessible to other users.

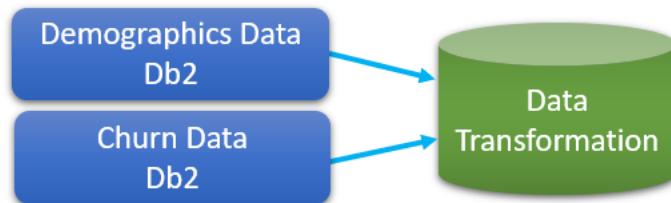
For the sake of simplicity, you will remain logged in the CPD console throughout this workshop as the user [Admin](#), which has been granted all persona roles. This was done so that you will not be required to log off and log on again as different users to represent the varying personas as you make your way through each lab in this workshop.

In your organization however, it is likely that once you have a mature CPD environment set up, separation of duties will be defined by persona where different users will be assigned one or more personas to do their particular tasks.

3.3 Db2 data overview – Transforming for analytics

The Db2 data in our scenario (Demographics and Churn) have two key factors that make an appropriate scenario for [Data Transformation](#):

- The data only changes once a month (it is relatively static). Thus, copying it and/or changing it for downstream processing in our CPD analytics workflow is OK because we are not required to have the absolutely latest data to get the results we need.
- The data has to be changed before it can be processed in our CPD analytics workflow.



Although you will be doing the actual [Data Transformation](#) in the next lab, you will prepare for it in this one by doing some [Collect](#) steps.

3.3.1 Review the Db2 data

- 42. To review this data, you will return to the *Db2 Data Server Manager* (DSM) console that you encountered in our first lab.

Start at the [Navigation Menu](#)



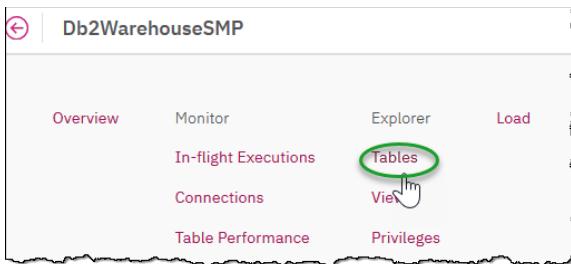
Click [Collect](#) ⇔ [My Data](#) ⇔ [Databases](#) ⇔ [Db2WarehouseSMP](#) ⇔ ellipsis ⋮ ⇔ [Open](#)

A screenshot of the 'My data' interface in the IBM Cloud Private for Data DSM. The 'Databases' tab is selected. Three database entries are listed: 'MongoDB-Activity1' (MongoDB Enterprise, Available), 'MongoDB-Activity2' (MongoDB Enterprise, Available), and 'Db2WarehouseSMP' (Db2 Warehouse SMP, Available). A context menu is open over the 'Db2WarehouseSMP' entry, with several options highlighted with green checkmarks: 'Details', 'Open', 'Configure', 'Submit connection for approval', and 'Access management'. The 'Delete' option is also visible at the bottom of the menu.

- __43. At the top right of the Db2 DSM console, click on the hamburger **Menu** icon.



Then choose the option **Tables**



- __44. Select schema **DFD**



- __45. Select table **CUSTOMER_CHURN** and click on it to bring up the table definitions view.

Then click on **View Data**

ID	CHURNRISK
0	Low
1	Low
2	Low
3	High
4	High
5	High

DFD.CUSTOMER_CHURN	
ID	CHURNRISK
0	Low
1	Low
2	Low
3	High
4	High
5	High

- __46. Click on the [Back](#) icon to view the data for table **CUSTOMER_DEMOGRAPHICS**

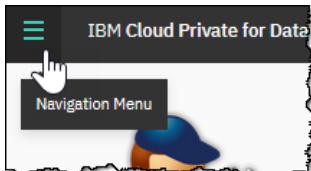
The screenshot shows the IBM Cloud Private for Data interface. On the left, there's a sidebar with a 'Tables' section containing a green checkmark icon and a 'Back' button. The main area has two tabs: 'Schemas' and 'Tables'. Under 'Schemas', 'DFD' is selected, showing '2 tables'. Under 'Tables', 'CUSTOMER_DEMOGRAPHICS' is selected, showing its data. A green arrow points from the 'Back' button in the sidebar to the 'CUSTOMER_DEMOGRAPHICS' table in the main area.

ID	GENDER	STATUS	CHILDREN	ESTINCOME	HOMEOWNER	AGE
0	F	S	1	38000.00	N	24
1	M	M	2	29616.00	N	49
2	M	M	0	19732.80	N	51
3	M	S	2	96.33	N	56

3.3.2 Review the Db2 connection

- __47. Review the connection to the Db2 Warehouse that we updated in our first lab.

Start at the [Navigation Menu](#)



- __48. Click [Connections](#) ⇒ [Db2WarehouseSMP](#).

Click on the connection to review it.

The screenshot shows the 'Edit connection' dialog for the 'Db2WarehouseSMP' connection. The connection name is 'Db2WarehouseSMP'. The JDBC URL is 'jdbc:db2://192.168.142.21:32267/BLUDB'. The connection type is 'Db2'. There are options for 'Use SSL' and 'Verify server SSL certificate'. An optional 'SSL certificate' field is present.

3.3.3 Add the Db2 datasets

_49. Review the connection to the Db2 Warehouse that we updated in our first lab.

Start at the [Navigation Menu](#)



_50. Click [Collect](#) ⇒ [My data](#)

From the [Data set](#) tab, click [+ Add new data set](#)

A screenshot of the "My data" page. At the top left is a back arrow icon. Next to it is the text "My data" followed by a green checkmark icon. Below this is a horizontal navigation bar with three tabs: "Data set" (which is highlighted with a red underline), "Data source", and "Databases". Underneath the tabs is a section titled "Data sets" with a green checkmark icon next to it. To the right of the title is a button labeled "+ Add new data set" with a green checkmark icon above it. Below this section is a table with two columns: "Project" and "Name". There is one row in the table. The "Project" column contains the value "1" and the "Name" column contains the value "CustomerOfferAccepted".

_51. From [Select a project](#), choose [TradingCustomerChurn](#)

A screenshot of a "Select a project" dialog. At the top left is a green checkmark icon. Next to it is the text "Select a project" followed by a red underline. To the right are two other options: "Local File" and "Remote Data Set". Below this is a list of projects. The first item in the list is "Select an analytics project" with a radio button next to it. The second item is "Create a new analytics project" with a radio button next to it. Below this list is a search bar containing the text "TradingCustomerChurn". To the right of the search bar is a green checkmark icon.

__52. Click the tab **Remote Data Set** and fill in like this:

Select a source: Db2Warehouse
Remote data set name: CUSTOMER_CHURN_TABLE
Description: CUSTOMER_CHURN table in DFD schema of the Datawarehouse
SQL object type: Table
Schema: DFD
Table: CUSTOMER_CHURN

Remote Data Set

Db2Warehouse

CUSTOMER_CHURN_TABLE

CUSTOMER_CHURN table in DFD schema of the Datawarehouse

Table

DFD

CUSTOMER_CHURN

__53. Click **Save**



__54. The dataset is added, now add another.

Click **+ Add new data set**

- __55. Click the tab **Remote Data Set** and fill in like this:

Note: Please name the table exactly as shown before saving.

Select a source:

Db2Warehouse

Remote data set name:

CUSTOMER_DEMOGRAPHICS_TABLE

Description:

CUSTOMER_DEMOGRAPHICS table in DFD schema

SQL object type:

Table

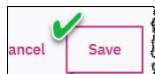
Schema:

DFD

Table:

CUSTOMER_DEMOGRAPHICS

- __56. Click **Save**



- __57. The **My data** screen now shows your new data sets.

	Project	Name	Type	Size	Last Modified	Data source
1	TradingCustomerChurn	CUSTOMER_DEMOGRAPHIC_TABLE	table	-	16 May 2019, 9:45 PM	Db2Warehouse
2	TradingCustomerChurn	CUSTOMER_CHURN_TABLE	table	-	16 May 2019, 9:40 PM	Db2Warehouse
3	TradingCustomerChurn	CustomerOfferAccepted	table	-	14 May 2019, 4:22 PM	Db2Warehouse
4	TradingCustomerChurn	CustomerMerged	table	-	14 May 2019, 4:21 PM	Db2Warehouse
5	TradingCustomerChurn	TraderDataFinal.csv	CSV	12.46 KB	14 May 2019, 3:38 PM	Local File
6	TradingCustomerChurn	TraderData.csv	CSV	8.41 KB	14 May 2019, 3:38 PM	Local File

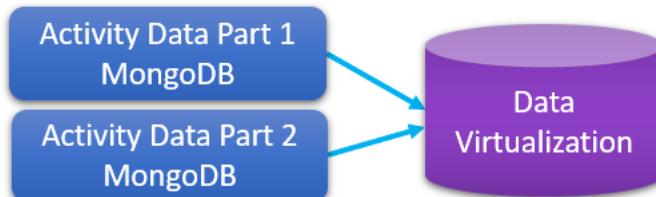
Now that we have these data sets defined, we could use them anywhere in our CPD downstream processing.

 Data Engineer	<p>The Data Engineer frequently works with connections and data sets to many different types of data sources.</p>
---	--

3.4 MongoDB data overview – Virtualizing for analytics

The Mongo data in our scenario (Activity) has two key factors that makes for an appropriate scenario for [Data Virtualization](#):

- The data is constantly changing (it is dynamic)
- Our analytics processing requires the absolute latest data



We have chosen semi-structured data in this scenario to demonstrate the power of CPD [Data Virtualization](#) – it even works with JSON documents. But we could just have easily chosen a structured database source for it as well, like Db2 or Oracle, SQL Server, and so on.

Semi-structured (and even unstructured) data is commonly used in many systems of engagement applications, so this was another reason we chose this as an example data source type. Our scenario presumes that this data comes from a mobile application that will be constantly changing the data, and we require the latest for our analytics workflow.

3.4.1 Review the MongoDB data

The Mongo databases were provisioned after the MongoDB Enterprise add-on was installed, which are also located in [My data](#).

__58. Start at the [Navigation Menu](#)

Click [Collect](#) ⇒ [My data](#) ⇒ [Databases](#) ⇒ [MongoDB-Activity1](#) ⇒ ellipsis ⋮ ⇒ [Open](#)

The screenshot shows the 'My data' interface with the 'Databases' tab selected. A database named 'MongoDB-Activity1' is listed. An arrow points from the 'Open' button in the database card to the 'Details' link in the MongoDB Ops Manager login screen. A callout box notes that the Details screen has the password.



The very first time you log into this *MongoDB Ops Manager* console it will ask you for user and password credentials. We have saved these for you in a browser setting. However, should you encounter this screen, you can find the password by going back and reviewing the [Details](#) option, shown in the screenshot above. Simply copy and paste the password from the [Details](#) screen to perform this console login. By the way, the user is always [admin](#).

__59. In the *MongoDB Ops Manager* console, click [Deployment](#) on the top left of the screen.

Then in the [Processes](#) tab, under [TOPOLOGY](#) section, click on the [Replica Set](#) link

The screenshot shows the 'Deployment' screen of the MongoDB Ops Manager. The 'Processes' tab is selected. A callout box highlights the 'Replica Set' link under the 'TOPOLOGY' section of the 'mongo-1558399132-replica-set' card.

- __60. Click on the [Data](#) tab to review the JSON documents in database [mongodb](#), collection [traderinfo](#)

mongodb-1558399132-replica-set

Overview Real Time Metrics Data Performance Advisor

1 DATABASES 1 COLLECTIONS

+ Create Database

NAMESPACES

mongodb traderinfo

COLLECTION SIZE 283.43KB TOTAL DOCUMENTS 1000 INDEXES TOTAL

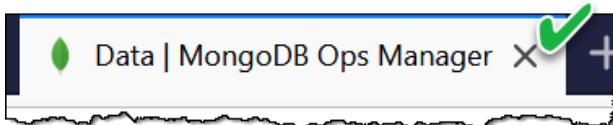
Find Indexes

FILTER {"filter": "example"}

QUERY RESULTS 1-20 OF MANY

_id: ObjectId("5ce3565e3fe54807bbccbbc2")
ID: 3
TotalDollarValueTraded: 26132.61
TotalUnitsTraded: 32
LargestSingleTransaction: 13066.305
SmallestSingleTransaction: 1500.0505
PercentChurnCalculation: 8

- __61. Close the *MongoDB Ops Manager* tab when finished

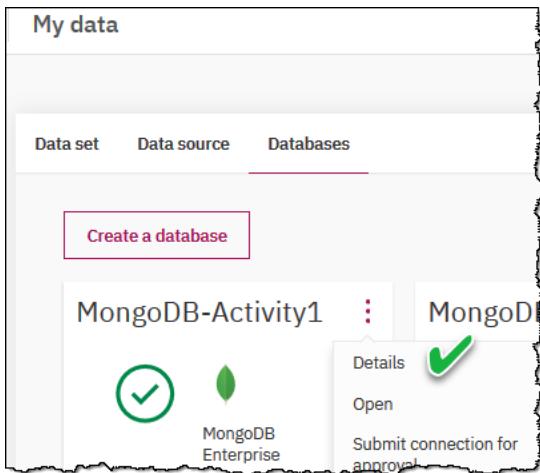


- __62. Do the same and review the data in the [MongoDB-Activity2](#) database

3.4.2 Create the MongoDB connections

Now let's create connections for the MongoDB databases to be used in our CPD workflow. Before creating a connection, get the information you need for it from the Details screen for that database.

63. In [My data](#), click on the [Details](#) of the database [MongoDB-Activity1](#)



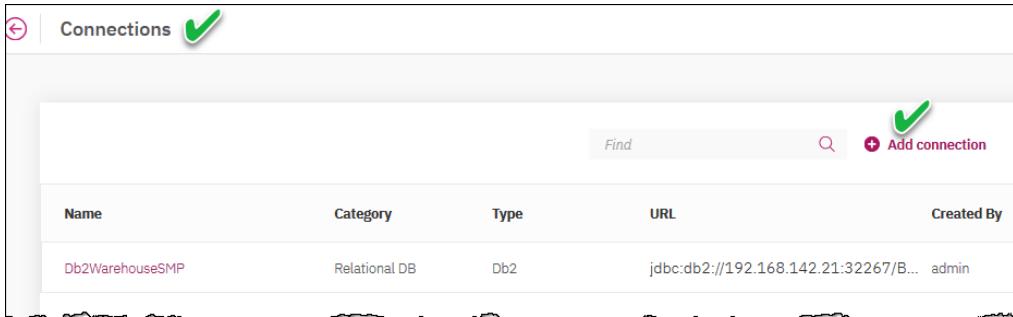
Make note of the host name - write it down (note: the port is always [27017](#))

Click copy next to the password (note: the user is always [admin](#))

The screenshot shows the 'MongoDB Ops Manager' details screen. It includes sections for 'HOSTNAME', 'CPU', and 'MEMORY'. Under 'HOSTNAME', the value '10.1.135.44' is highlighted with a green callout bubble containing the text 'write this down'. Below this, there are sections for 'Storage class' (oketi-gluster) and 'Size' (100 GB). The 'MongoDB Ops Manager' section shows 'First Ops Manager user' as 'admin' and 'Password' as 'iDdNC%6313a%*Q@3n'. A green callout bubble next to the password field contains the text 'click to copy this'.

_64. Next start at the Navigation Menu

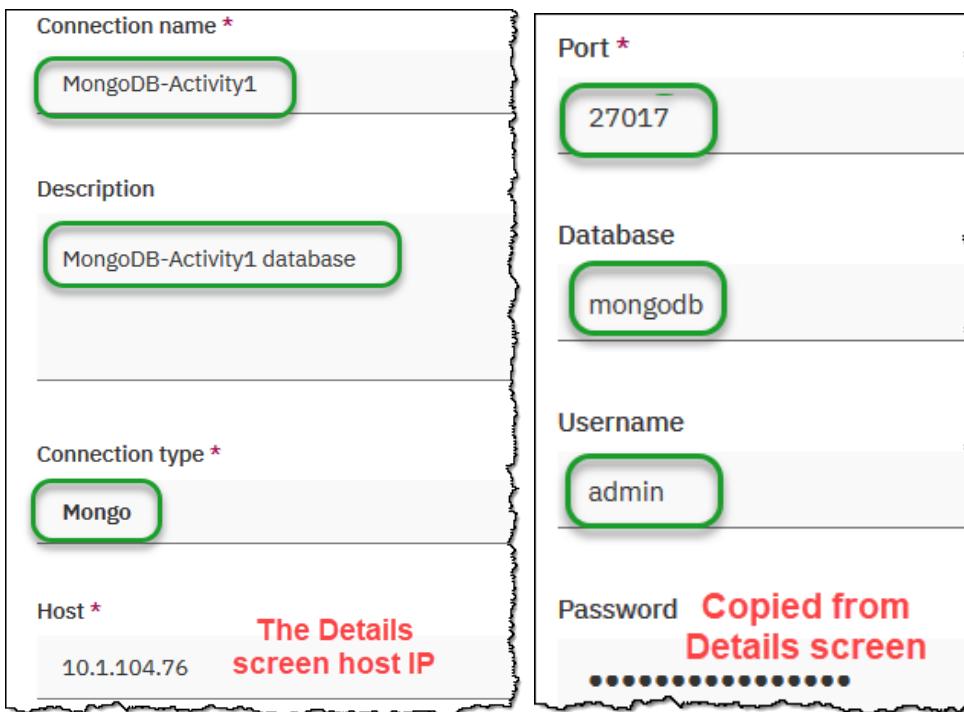
Click **Connections** \Rightarrow **+ Add connection**



Name	Category	Type	URL	Created By
Db2WarehouseSMP	Relational DB	Db2	jdbc:db2://192.168.142.21:32267/B...	admin

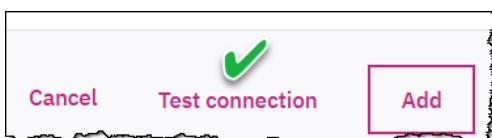
_65. Fill in the details of this screen like this:

Connection name: **MongoDB-Activity1**
 Description: **MongoDB-Activity1 database**
 Connection type: **Mongo**
 Host: Host IP you wrote down from the Details screen of the database
 Port: **27017**
 Database: **mongodb**
 Username: **admin**
 Password: Copied from the Details screen of the database

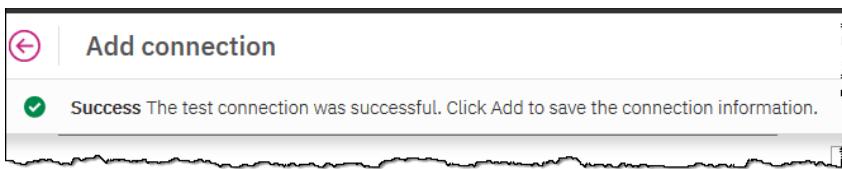


Connection name *	MongoDB-Activity1
Description	MongoDB-Activity1 database
Connection type *	Mongo
Host *	10.1.104.76 The Details screen host IP
Port *	27017
Database	mongodb
Username	admin
Password	Copied from Details screen [REDACTED]

_66. From bottom of this connection edit screen, **Test** the connection:

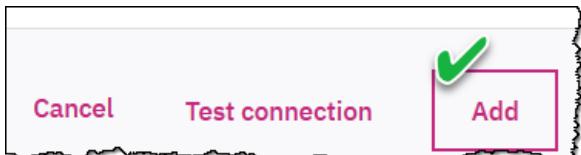


- _67. The top of the screen should show a successful test.



 Data Engineer	If your test is not successful, check all the credentials again, especially Host and Password.
---	--

- _68. After a successful connection test, click **Add**



- _69. Your **Connections** screen now shows a new connection with the Type: Mongo

Connections				
Name	Category	Type	URL	Created By
MongoDB-Activity1	NoSQL DB	Mongo	jdbc:ibm:mongodb://10.1.104...	admin
Db2WarehouseSMP	Relational DB	Db2	jdbc:db2://192.168.142.21:3...	admin

- _70. Now create a connection for database **MongoDB-Activity2** making sure to get the correct **Host** and **Password** from the **Details** page for that database. (Hint: **My Data** \Rightarrow **Databases**)

- _71. When you are done, click the twistie in **Connections** to sort them.

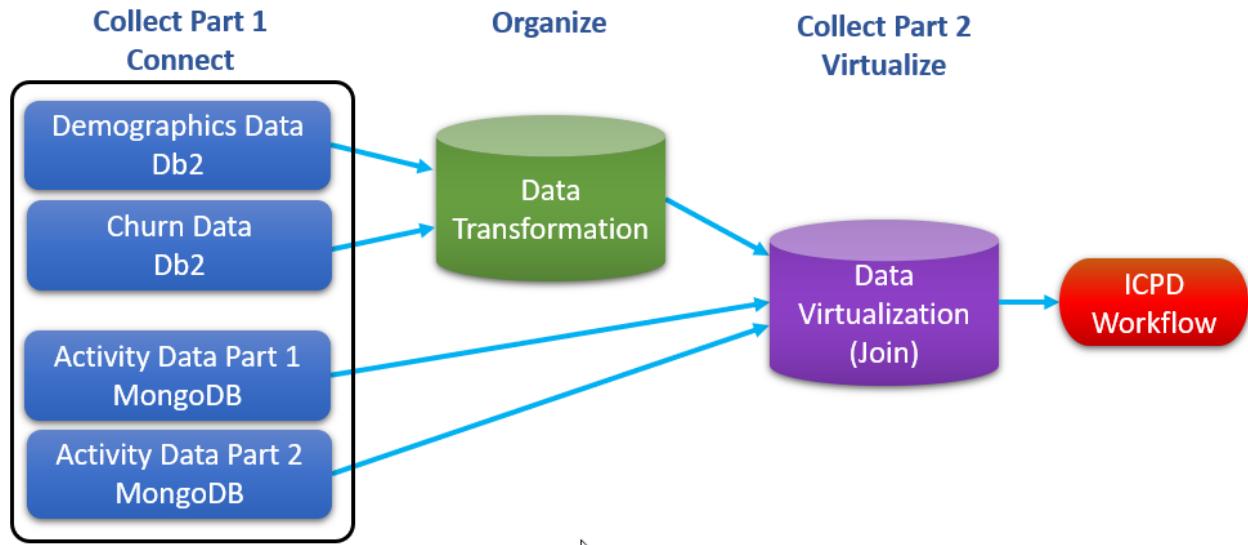
Make sure you created two different MongoDB connections and that you did not accidentally create two connections for the same MongoDB database. The host/port combination should be different for each.

Name	Category	Type	URL	Created By
Db2WarehouseSMP	Relational DB	Db2	jdbc:db2://192.168.142.22:32042/BLUDB	admin
MongoDB-Activity1	NoSQL DB	Mongo	jdbc:ibm:mongodb://10.1.135.44:27017;dat...	admin
MongoDB-Activity2	NoSQL DB	Mongo	jdbc:ibm:mongodb://10.1.135.3:27017;dat...	admin

Make sure
these are
unique

3.5 Lab conclusion

In this Collect Part 1 lab, you created connections for the Db2 and MongoDB data. This prepares you for the Organize and Collect Part 2 Virtualize steps in our workshop.



In the [Organize](#) lab coming up, we will be transforming the Db2 data sets into one.

After that, we will finish up the [Collect](#) processing by virtualizing the results from both the Db2 Transformation output and the MongoDB data together.

**** End of Lab 03: Collect Part 1 - Connect**

Lab 04 Organize

4.1 Lab overview

Many organizations find it difficult to understand their own data because it originates from many sources, is dispersed across many silos, and is controlled by different teams.

This [Organize](#) lab will show you how to uncover the hidden data in your organization's data and how to build a lineage that is otherwise difficult to establish. Cloud Pak for Data helps your organization move from the manual processes required to establish relationships between data, to an automated one aided by the platform's built-in machine learning capabilities.

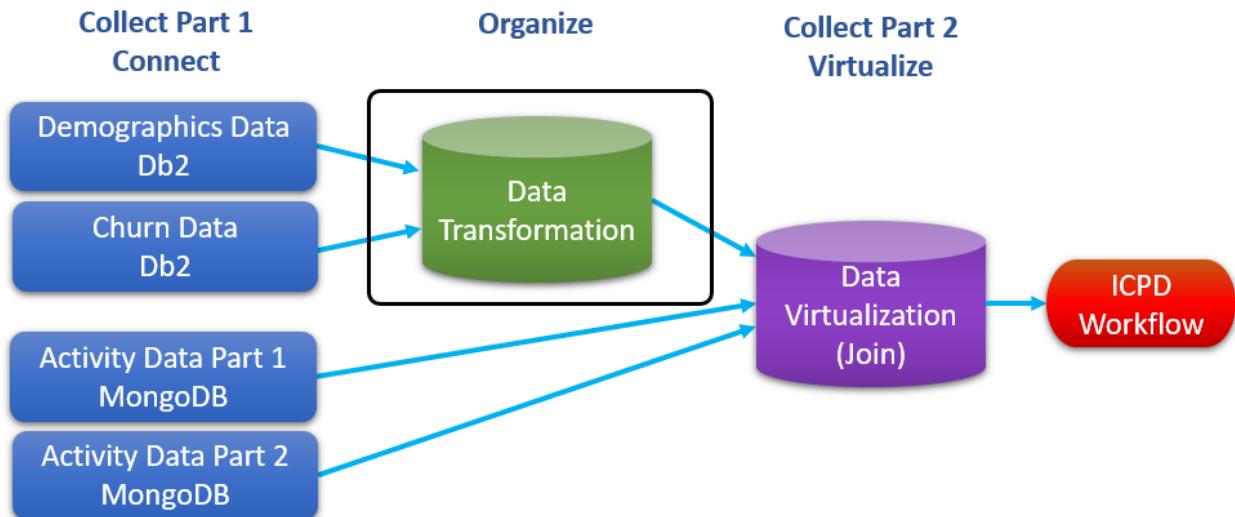
4.2 Persona represented in this lab

The [Data Steward](#) persona is the likely one to perform most of the [Organize](#) tasks shown in this lab.

Persona (Role)	Capabilities
 Data Steward	Data Stewards brings integration and governance to the data.

Note that this persona often works closely with the [Data Engineer](#) persona because they both work with the data to prepare it for the analytics processing by other personas.

For example, in this lab one of things the [Data Steward](#) will do is to create a Transformation job with the Db2 data. In turn, the [Data Engineer](#) will then use that file to create a final virtualized table of all the data sources joined together.



Before we start transforming data, let's first explore the other crucial aspects of the CPD [Organize](#) capabilities, starting with creating a Business Glossary.

4.3 Create a business glossary

A **business glossary** consists of **categories** and **terms**.

Categories provide the logical structure for the glossary so that you can browse and understand the relationships among terms and categories in the glossary. Categories can be organized in a hierarchy based on their meaning and relationships to one another.

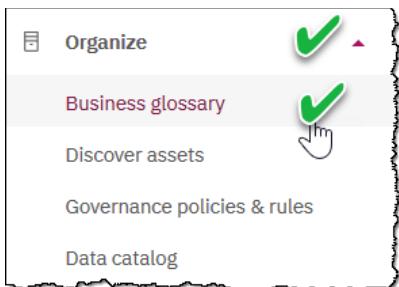
A **term** is a word or phrase that describes a characteristic of the enterprise. Terms are the fundamental building blocks of the glossary. Each term has a parent category, but it can also be referenced by other categories. When you create a term, you need to provide a meaningful name. Terms can be assigned to other terms, and to other asset types as well.

Let's explore building a business glossary in CPD.

- __72. Start at the **Navigation Menu**



- __73. Click **Organize** ⇒ **Business Glossary**



- __74. Under the tab **Categories** ⇒ **Create Category**.

NAME	CREATED	MODIFIED
Customer Churn	6 May 2019, 2:58 AM	6 May 2019, 3:24 AM

- __75. Type **Name**: Customer Churn and **Short Description**: Terms related to the reason a customer stops doing business with our company

Create Category

Name*
Customer Churn 241

Parent Category
Type to find and add This is a Parent Category, so leave blank

Short Description
Terms related to the reason a customer stops doing business with our company 179

- __76. Click **Save** (bottom right of the screen)



- __77. Go to the second tab **Terms** and click **Create Term**.

Categories Terms ✓

Search for terms in the catalog

7 terms available Import Terms Create Term ✓

- __78. Create a term **Income** with a parent category **Customer Churn** (hint: type **C** in the box), provide a description as: **Yearly income of the customer** and choose **Status** as **Standard** from the drop-down menu.

Name*
Income 249

Parent Category*
Customer Churn

Short Description
Yearly income of the customer 225

Status*
Standard ✓

- __79. Click the down arrow next to **Save** and click **Save and Create another Term**.



- __80. Now, repeat above exercise to create the following **Terms** in the table below.

After each use **Save and Create another Term**, except for the last which you **Save**.

Name	Parent Category	Short Description	Status
Home Owner	Customer Churn	Flag indicating whether the customer owns a home.	Standard
Net Realized Losses	Customer Churn	The net dollar value of losses realized for a customer, usually a year to date metric	Standard
Net Realized Gains	Customer Churn	The net realized gains for a customer, usually a year to date metric.	Standard
Gender	Customer Churn	The gender of the customer, either 'M' or 'F'	Standard
Days Since Last Trade	Customer Churn	The number of days since the customer last executed a trade.	Standard
Total Dollar Value Traded	Customer Churn	The total amount the customer has traded with us since onboarding.	Standard

- __81. Review your new Business Glossary **Category** and **Terms** related to that category

NAME	CREATED
Income	18 May 2019, 2:29 PM
Net Realized Losses	18 May 2019, 2:30 PM

 Data Steward	<p>You can create your own glossary with categories and terms manually, or import them from a file. In addition, you can import Industry Models from IBM for industries such as finance, banking, healthcare, and insurance and import them into CPD.</p> <p>See the add-ons screen  then Industry Accelerators.</p>
--	--

4.4 Create governance policies and rules

An information governance **policy** is a natural-language description of a governance subject area. It can contain multiple information governance sub-policies or reference one or more information governance rules. It must fulfill a business objective, and be relevant and understandable to all users of the policy.

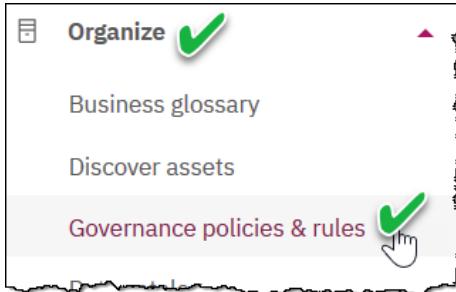
An information governance **rule** is a natural-language description of the criteria that are used to determine whether information assets are compliant with business objectives. Generally, information governance rules are derived from information governance policies and are more specific. The rules define the actions to take in specific situations to implement the policy. An information governance rule can be referenced by one or more information governance policies.

Let's explore creating these in CPD.

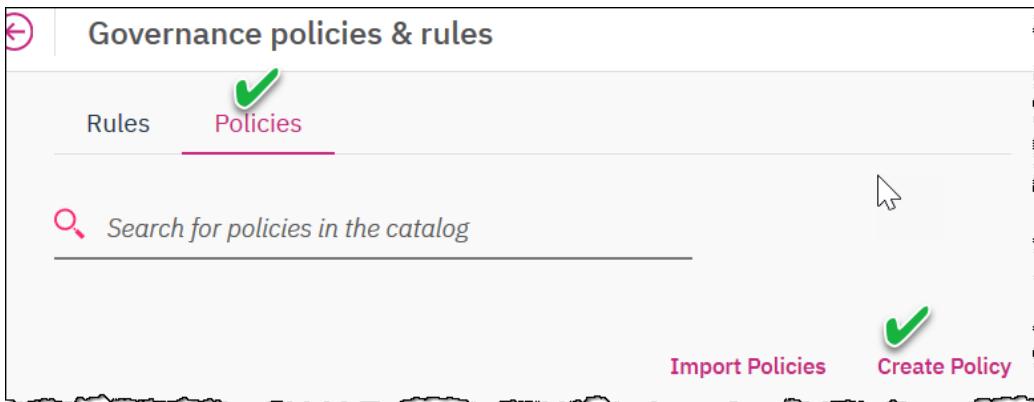
- _82. Start at the [Navigation Menu](#)



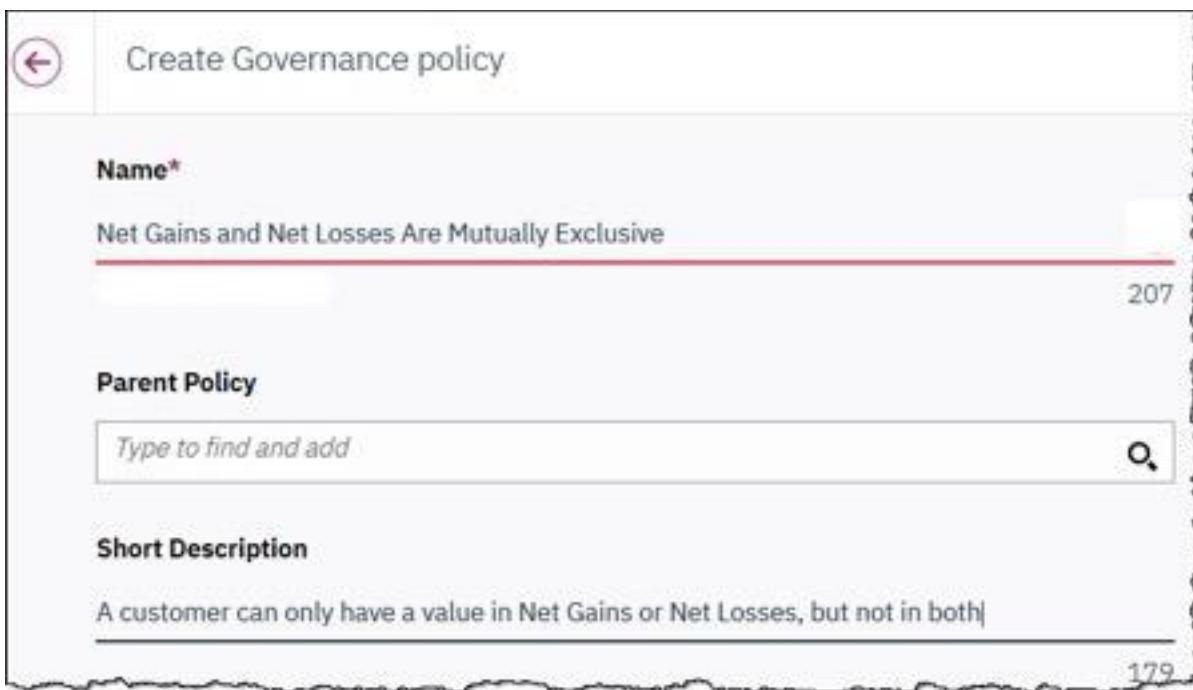
- _83. Click [Organize](#) ⇒ [Governance policies and rules](#)



- _84. You will start by creating a Governance policy. Click the tab [Policies](#) ⇒ [Create Policy](#).



- 85. Type **Name**: Net Gains and Net Losses Are Mutually Exclusive and **Description**: A customer can only have a value in Net Gains or Net Losses, but not in both.



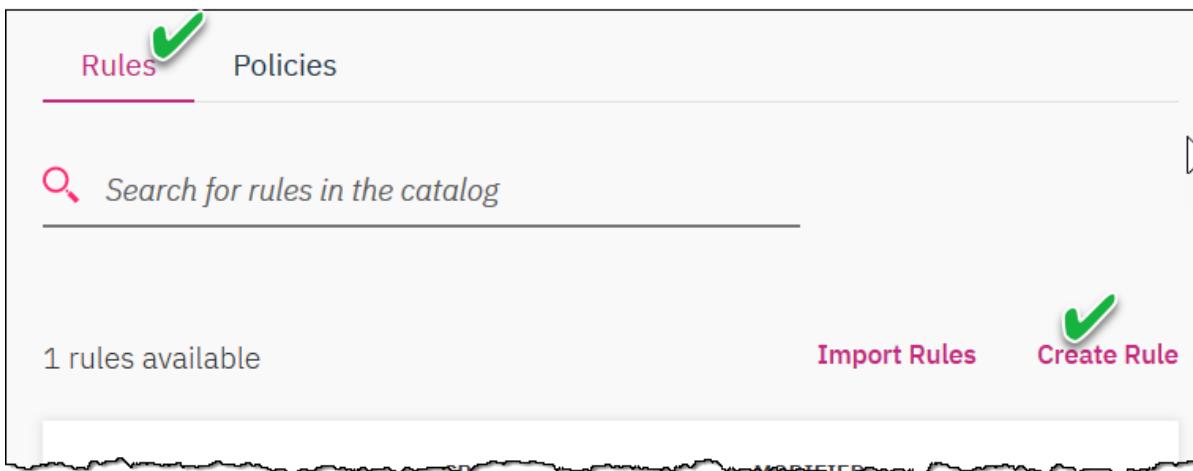
Create Governance policy

Name*
Net Gains and Net Losses Are Mutually Exclusive
207

Parent Policy
Type to find and add

Short Description
A customer can only have a value in Net Gains or Net Losses, but not in both|
179

- 86. Click **Save**.
—87. We just created a policy. Now create a rule for it.
Click tab **Rules** ⇒ **Create Rule**.



Rules Policies

Search for rules in the catalog

1 rules available

Import Rules

Create Rule

__88. Type Name: **Net Realized Gains and Losses Validity Check**,

Type '**n**' to select the Referencing Policy

Type Description: **If Net Realized gains > 0 then Net Realized Losses = 0 else if Net Realized Losses > 0 then Net Realized Gains = 0.**

Create Governance rule

Name*
Net Realized Gains and Losses Validity Check 211

Referencing Policies
Net Gains and Net Losses Are Mutually Exclusive X Q

Short Description
If Net Realized gains > 0 then Net Realized Losses = 0 else if Net Realized Losses > 0 the 141

__89. Click **Save**.

 Data Steward	A Data Dictionary contains a business glossary (categories and terms) and information governance policies and rules to ensure data compliance with business objectives. In this lab, with the creation of your 1 category and 7 terms, 1 policy and 1 rule, you could now say the organization's Data Dictionary has a nice start. In reality, a Data Dictionary for any organization can and should be updated as frequently as new data sources, regulations and other criteria require it.
---	--

4.5 Discover assets

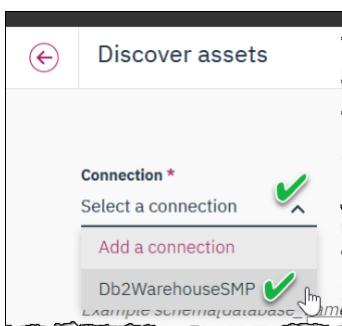
[Discover assets](#) allows you to do three things: 1) Profile and classify data, 2) Analyze the quality of data, 3) Assign business terms to data.

Let's see how this works.

- __90. Start at the [Navigation Menu](#)
- __91. Click twistie [Organize](#) and expand it and click [Discover assets](#).



- __92. Click on the [Connection](#) dropdown arrow, then select [Db2WarehouseSMP](#)



- __93. In [Schemas](#), click [Browse](#) (Note: Choose Db2Warehouse if the SMP is not there.)



- __94. Expand dropdown [db2](#) and select schema [DFD](#) ONLY, then [OK](#)



___95. The checked boxes below the **Schemas** field must be check in the following order:

- 1) Profile and classify data
- 2) Assign business terms
- 3) Analyze data quality
- 4) Use data sampling
- 5) Set the maximum number of records = 1,000
- 6) Select first button below maximum records field

The screenshot shows the 'Schemas' configuration screen. At the top, there is a 'Schemas' field containing 'schema[db2|DFD]'. Below it, a section titled 'Select the tasks that you want to run' contains six numbered options: 1) Profile and classify data (checked), 2) Assign business terms (unchecked), 3) Analyze data quality (checked), 4) Use data sampling (checked). Option 5) is a text input field containing '1000'. Option 6) is a radio button group with two options: 'Use the first x number of rows (where x = maximum number of records allowed)' (selected) and 'Use every Nth value (up to maximum number of records)'.

___96. Click **Discover**. (Bottom right of the screen.)



___97. A job is created and begins to execute. It may take few minutes to complete.



___98. You can click the **Refresh** icon to see if the job is still running.



___99. You can watch the progress of this job and refresh as needed. This should take 4 or 5 minutes.

DISCOVERED ASSETS INFORMATION		
Number of schemas 1		Number of tables 2
Asset name	Asset type	Status
DFD	Schema	Phase Import Finished Start May 20th 2019, 11:02:06 am End May 20th 2019, 11:02:19 am
		Phase Analyze Running Start May 20th 2019, 11:02:30 am Done 0% Successful 0% Cancelled 0% Failed 0%

Notice the Discovery phase has two stages – 1. [Import](#) and 2. [Analyze](#).

The Discovery process can take some time, so be patient while it completes.

Hit the refresh  button after a few minutes to see if it has completed.

Try again in another minute. (Note: If you get failure on analyze run analyze again.)

 Data Steward	<p>Occasionally the Discover process fails to finish after 10 minutes or so. If this happens to you, see Appendix B – Lab fixes – Lab 04 “Runaway Discover”.</p> <p>Note: a successful run of Discovery is NOT necessary to complete this lab, nor this workshop. If you wish to skip it, you may do so by going directly to the next section: Shop for Data.</p>
---	---

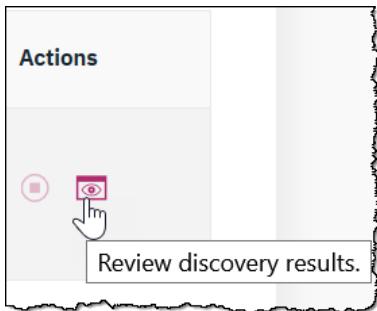
___100. When it is finished, you will see that the ‘Analyze’ phase has a flag marked [Finished](#).

Click the back arrow to return to the Discover screen.



 Data Steward	<p>This is an example of an incredibly powerful automated discovery, with an assignment of a quality score as well as Terms (if there was an appropriate match) so that it can be better used by the personas who may need it in their analytic processing.</p> <p>Simply by pointing to the data we want to process with CPD internal machine learning capabilities, CPD profiled and classified the data so that the business purpose is more easily understood. For example, it can discern PII data, the significance of some columns over others, and so on.</p>
---	---

__101. Now you can review the results by clicking on the [review discovery results](#) (eye) icon:



__102. Expand the two tables [CUSTOMER_CHURN](#) and [CUSTOMER_DEMOGRAPHICS](#)

Notice the data has been assigned a quality score at the table level as well as the individual table column level

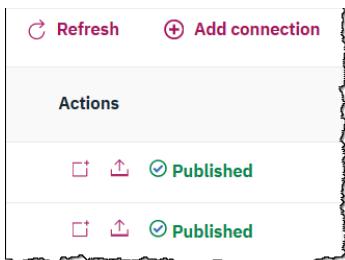
Notice too that column [Gender](#) was assigned the previously created term [Gender](#) with 100% certainty. Other columns were assigned terms with slightly less certainty.

Assets				
<input type="checkbox"/>	Name	Quality	Data class	Assigned terms ?
<input checked="" type="checkbox"/>	CUSTOMER_CHURN	91%	-	-
	CHURNRISK	81%	City 82% ▾	-
	ID	100%	Identifier 100% ▾	-
<input checked="" type="checkbox"/>	CUSTOMER_DEMOGRAPHICS	100%	-	-
	GENDER	100%	Gender 100% ▾	Gender 100% X
	HOMEOWNER	100%	Person Name 75% ▾	Homeowner 100% X
	AGE	100%	Code 100% ▾	-
	ID	100%	Identifier 100% ▾	-
	STATUS	100%	Organization Name 100% ▾	-

__103. Publish these two tables to the catalog by clicking on publish icons as shown below:

Assets					Audit	Refresh	Add connection
<input checked="" type="checkbox"/>	Name	Quality	Data class	Assigned terms ?	Actions		
<input checked="" type="checkbox"/>	CUSTOMER_CHURN	91%	-	-	<input type="checkbox"/> Published		
<input checked="" type="checkbox"/>	CUSTOMER_DEMOGRAPHICS	100%	-	-	<input type="checkbox"/> Published	Publish this dataset to catalog	

__104. After submitting for publication, the screen will show this:



__105. Once published, we can search for the data. Try it by clicking on the **Search** icon in the top bar.

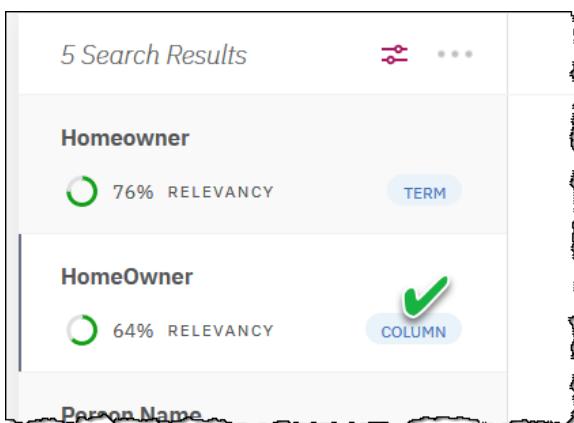


__106. Type **homeowner**

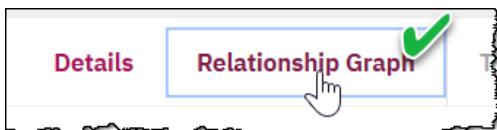


__107. Notice that a Term and a Column of a table come up.

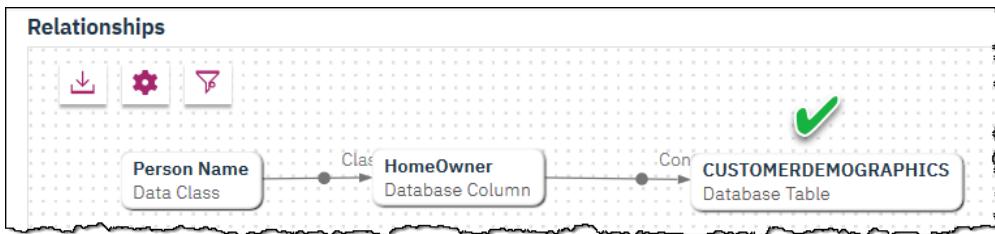
Click on the **COLUMN** called **HomeOwner**



__108. In the COLUMN details screen, click on **Relationship Graph**



- 109. After rearranging this screen, you will notice that this column belongs to the table CUSTOMERDEMOGRAPHICS. (Your illustration may vary)



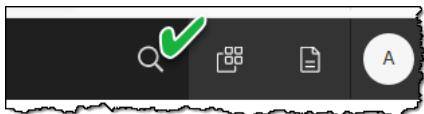
 Data Steward	<p>In our lab so far, you have only performed discover on a few tables so this relationship graph does not yet tell you of more complex relationships that it could know. You could do discovery for hundreds of databases to find relationships between them, as well as on your Data Dictionary items to uncover what Gartner calls “Dark Data” in your organization. This is data you are collecting, but not using because you do not know its value or how it relates to other data, or you cannot easily access it.</p>
--	---

4.6 Shop for Data

The Data Scientist and Business Analyst personas may not always know what has been made available to them by the Data Engineers and Data Stewards in CPD. In fact, individual Data Engineers and Data Stewards may not always know what other users of the same persona have made available through their Collect and Organize activities.

This is remedied by the ability to shop for data. You have already been introduced to the Search feature, which is the beginning step for shopping for data, so let's explore it further.

_110. Click the **Search** icon.



_111. Type **churn** and hit **Enter**.



_112. Assets related to **churn** are displayed.

Click the filter icon to more easily explore the results of this search.

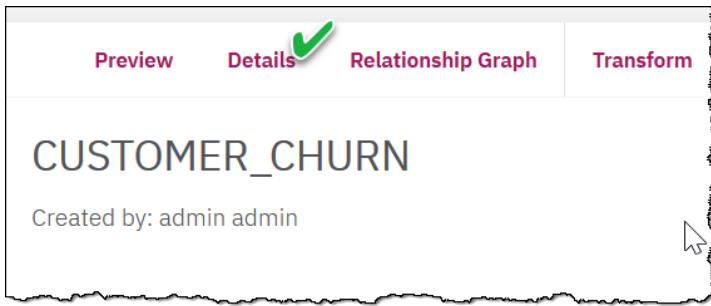
A screenshot of a search results interface titled "Search Results". It shows 16 search results. The first result is "CUSTOMER_CHURN" with 76% relevance and a "TABLE" button. The second result is "Customer Churn" with 72% relevance and a "CATEGORY" button. A green checkmark is placed over the filter icon at the top of the results list.

_113. Expand **Databases** then check **Database Tables**.

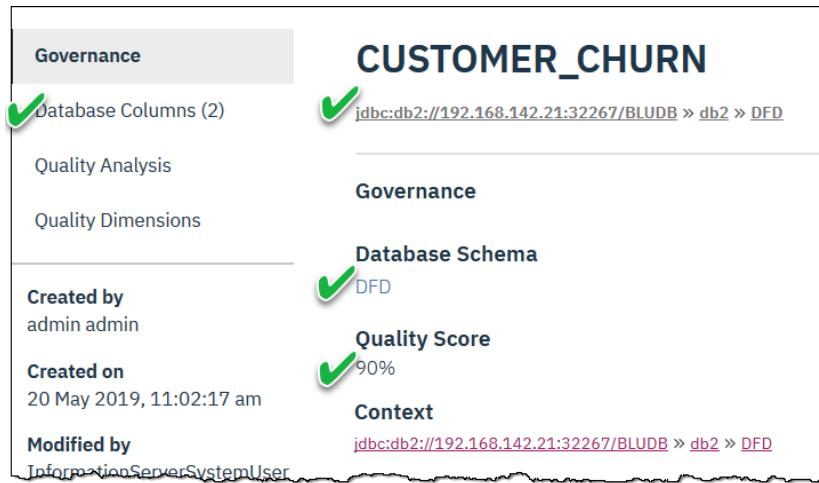
A screenshot of a navigation sidebar. The "Databases" section is expanded, showing "(1)" and a green checkmark. Below it are "Database Schemas" (1), "Database Tables" (2) which is checked and has a green checkmark, and "Database Columns" (3).

- Databases (1)
- Database Schemas (1)
- Database Tables (2)
- Database Columns (3)

_114. Click on **Details** to explore this table **CUSTOMER_CHURN**



_115. Review the lineage of this table, the database, schema, quality score, and number of columns.



_116. Return to the previous screen for the table **CUSTOMER_CHURN**



_117. Click **Preview**



__118. For the credentials to this database, enter *Username icpd, Password icpd*

Then click [Continue](#)

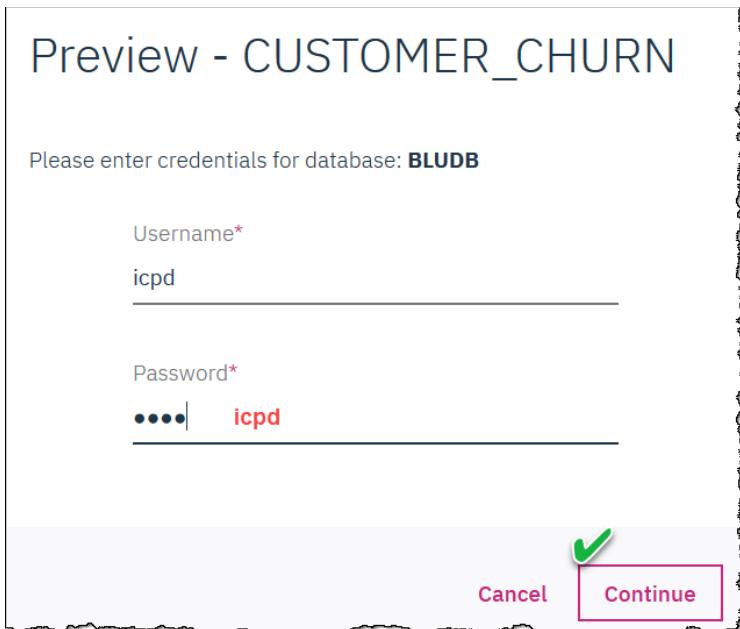
Preview - CUSTOMER_CHURN

Please enter credentials for database: **BLUDB**

Username*
icpd

Password*
•••• | **icpd**

 [Cancel](#) [Continue](#)



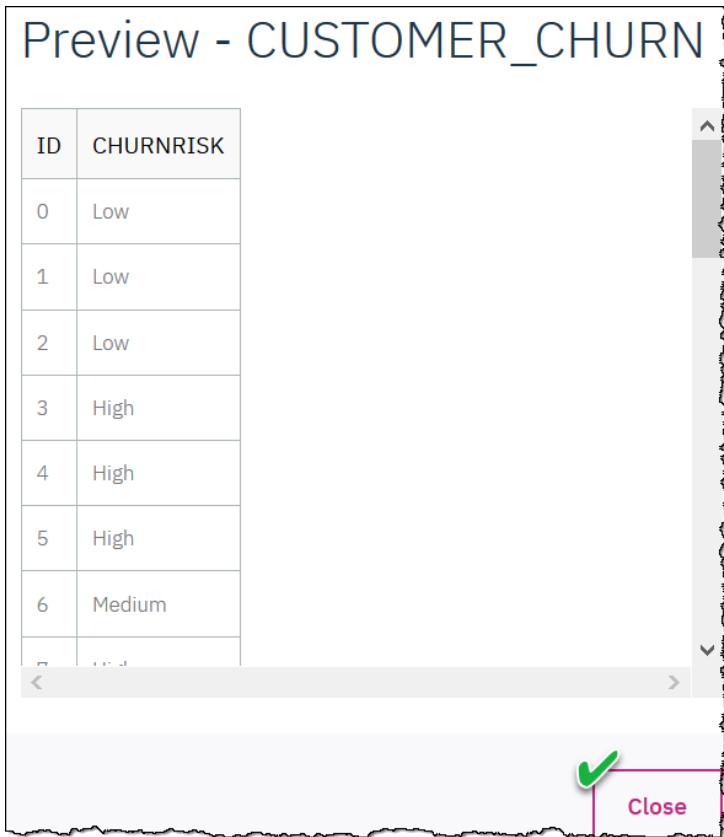
__119. Review the data itself to find out if it is suitable for your needs.

Then click [Close](#)

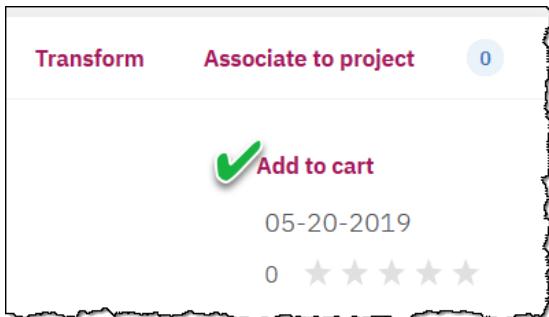
Preview - CUSTOMER_CHURN

ID	CHURNRISK
0	Low
1	Low
2	Low
3	High
4	High
5	High
6	Medium

 [Close](#)

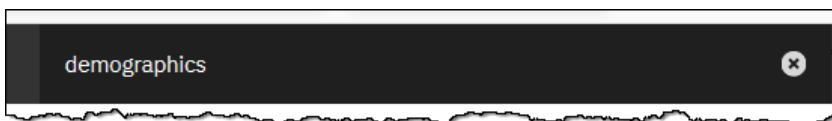


_120. The data looks good, so click [Add to Cart](#)



_121. You will need one more piece of data to complete your shopping experience.

The [Search](#) field should still be open – type [demographics](#) and hit [Enter](#)



Note: if you closed the [Search](#) field, just click on the icon again to start another search.

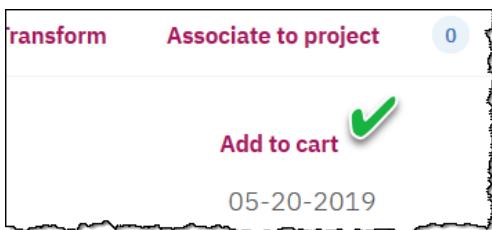


_122. The results show another table called [CUSTOMER_DEMOGRAPHICS](#). Click on it to choose it;



_123. To shorten this lab exercise, pretend you reviewed the details of this table and reviewed the data as well and that you have determined the data meets your needs.

Click [Add to cart](#)

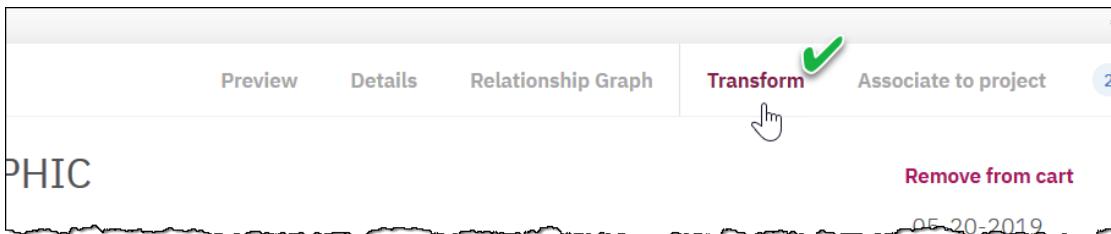


 Data Steward	<p>You have shopped for and chosen two tables for your cart, from which you will build a job to join and transform this data.</p>
--	---

4.7 Transform Data

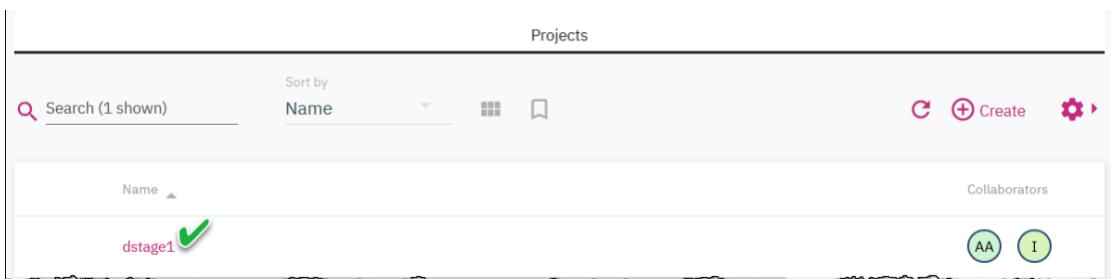
In this set of exercises, you will use the CPD built-in Data Flow Designer (DFD) to build a job that can join and transform your data.

- _124. Click **Transform**.



- _125. This is the DFD projects screen where an example is already listed.

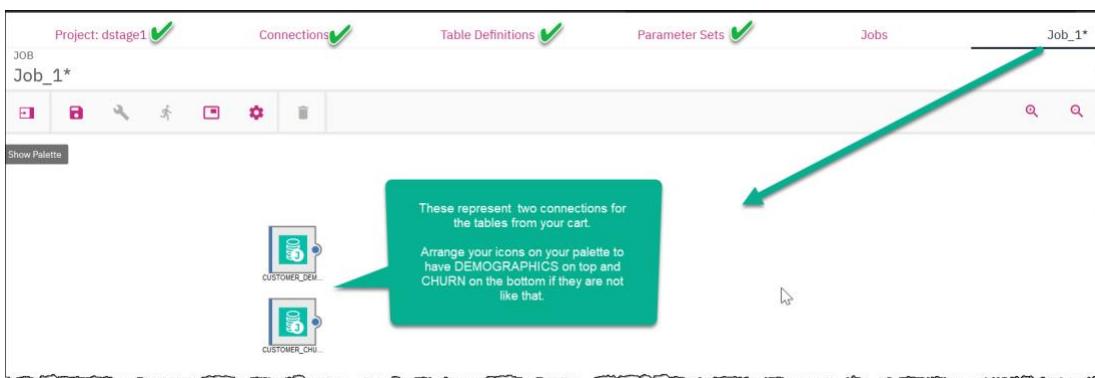
Click project **dstage1**.



- _126. The project **dstage1** canvas has across the top various options: **Connections**, **Table Definitions**, **Parameter Sets** and **Jobs**.

Notice DFD has created a new job for you called **Job_1*** and placed on the canvas the two connections for the data you added to your cart. These are for the CUSTOMER_DEMOGRAPHICS and CUSTOMER_CHURN tables.

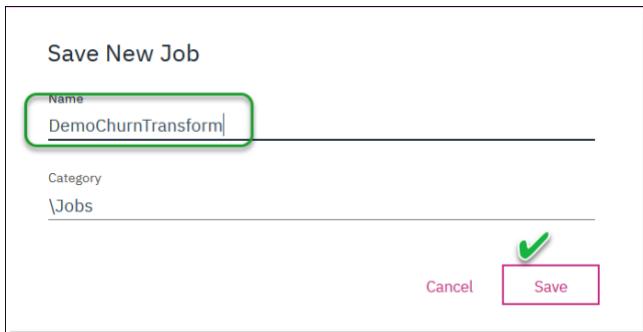
Arrange your icons to have DEMOGRAPHICS on top if they are not presented to you on your canvas that way. Note: If they do not link delete and re-add to cart.



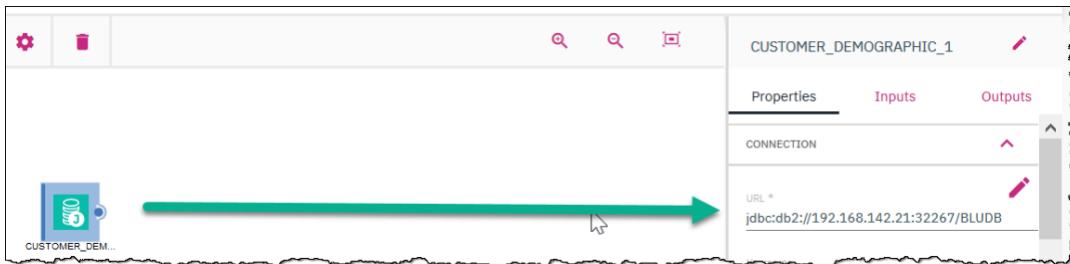
- _127. Save this job.



_128. Rename it to **DemoChurnTransform**



_129. Double click on the first connection on the canvas for **CUSTOMER_DEMOGRAPHICS**



_130. The pop up connection editor allows you to review and edit the characteristics of this connection.

Add the schema to the Table name, so it looks like this: **DFD.CUSTOMER_DEMOGRAPHICS**

Click **OK** to save and close this screen.

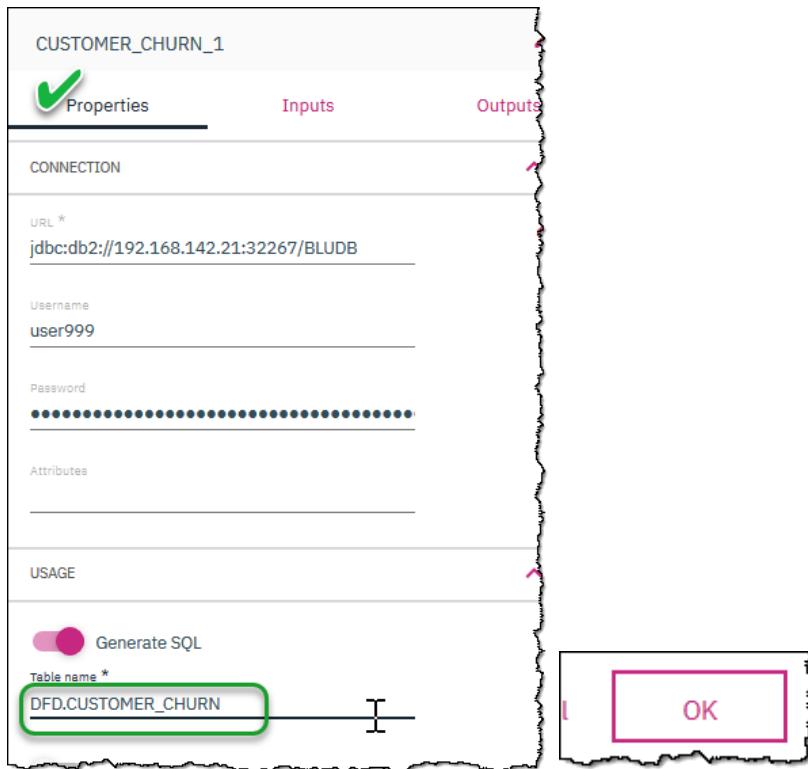
CUSTOMER_DEMOGRAPHICS_2	Properties	Inputs	Outputs
CONNECTION			
URL *	jdbc:db2://192.168.142.21:32267/BLUDB		
Username	user999		
Password	*****		
Attributes			
USAGE			
<input checked="" type="checkbox"/> Generate SQL Table name * <input type="text" value="DFD.CUSTOMER_DEMOGRAPHICS"/>			

Note: If this characteristics screen does not appear, use [CTL][+] and [CTL][-] to zoom out and in so that it will show.

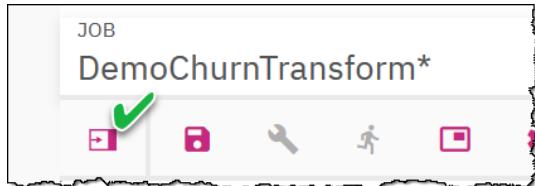
__131. Double click on the second connection icon on the canvas, which is CUSTOMER_CHURN

Add the schema to the Table name, so it looks like this: DFD.CUSTOMER_CHURN

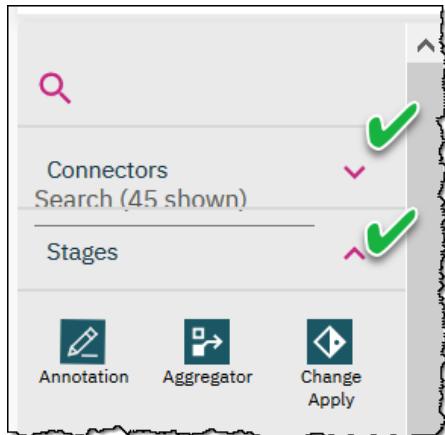
Click **OK** to save and close this screen..



__132. Review the palette options by clicking on the show palette icon:

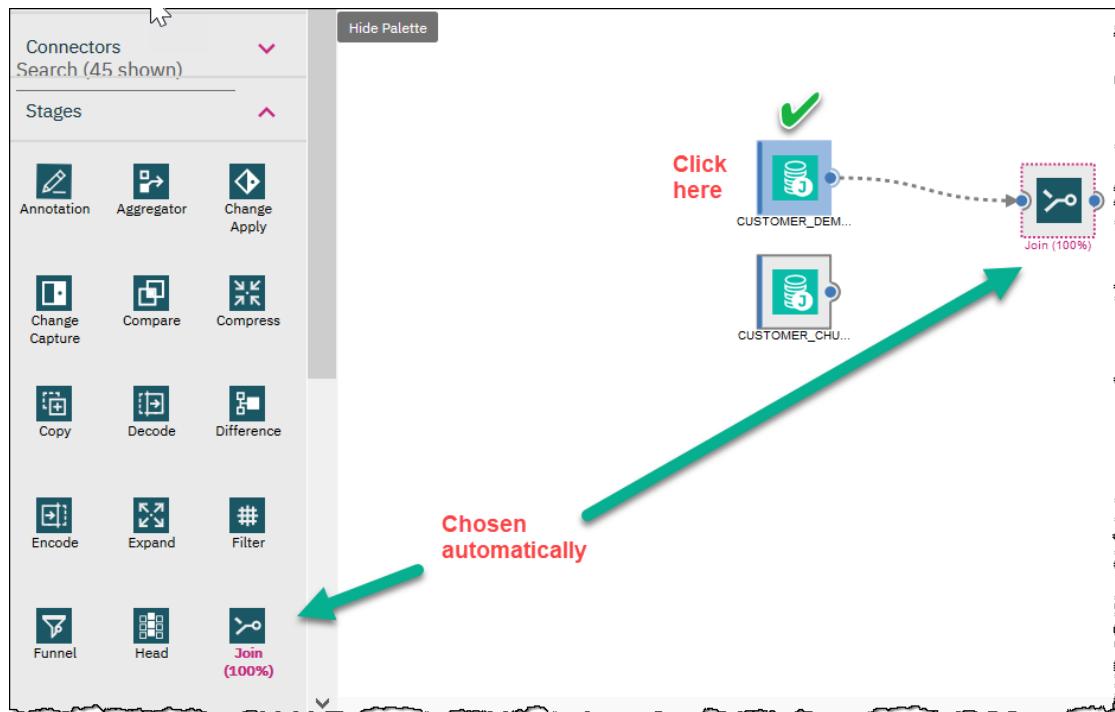


__133. Collapse the **Connectors** palette section and expand the **Stages** palette section:

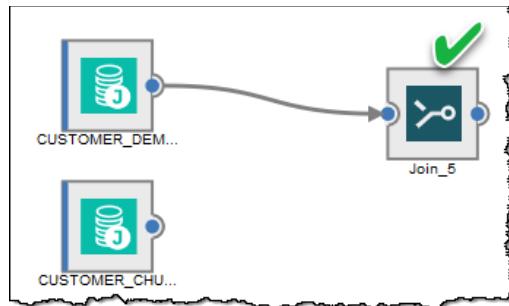


_134. Click once on the CUSTOMER_DEMOGRAPHICS connection icon on the canvas to choose it.

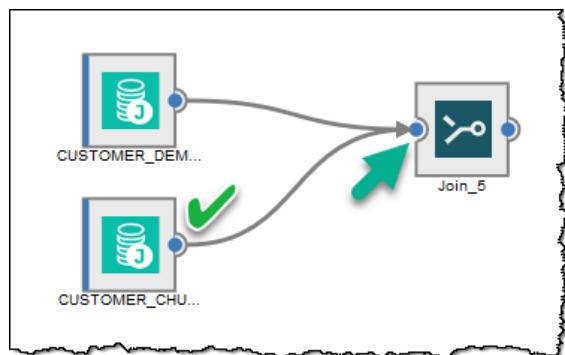
With the Stages palette options open, the Data Flow Designer does a “Smart Stage suggestion” picking the Join Stage from the palette as a possible next step.



_135. Click on the **Join** icon on the canvas to take the suggestion and make that choice. The icon will now have solid lines around it.



_136. Click on the nub of the CUSTOMER_CHURN connection icon and drag the connecting arrow to the **Join** icon. This provides the Join stage the two tables as sources for the join.

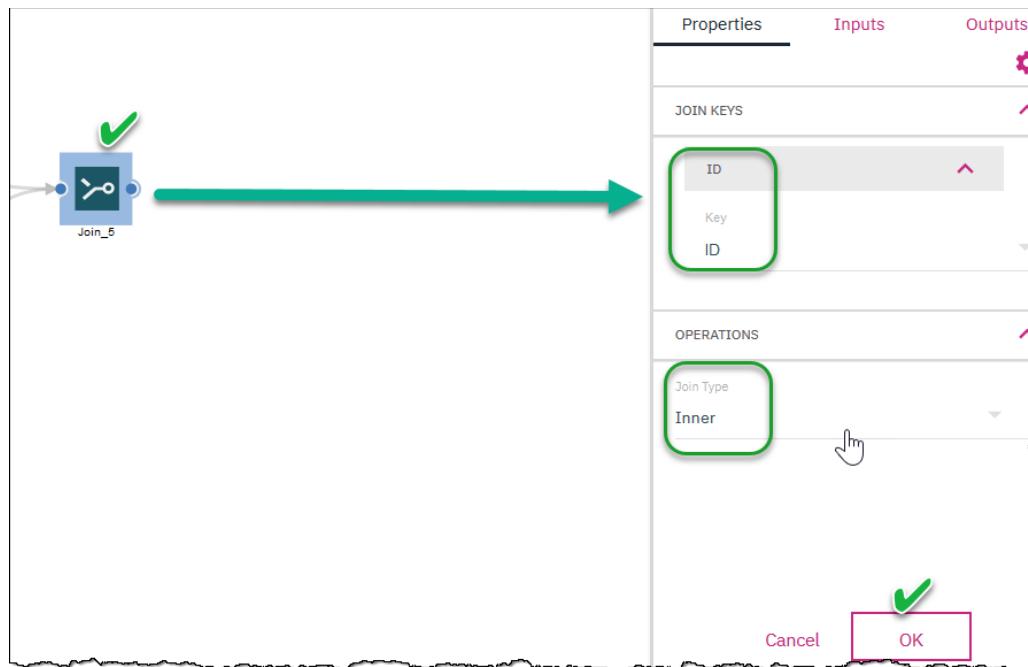


__137. Double-click on the Join stage icon to review its properties, which will pop up on the right.

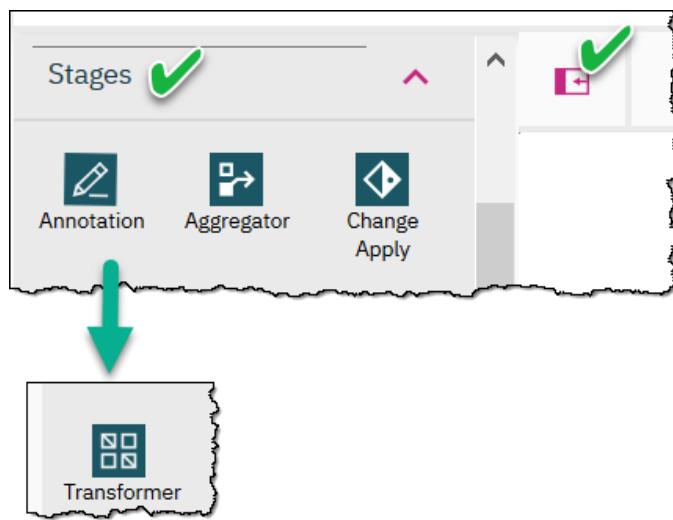
Data Flow Designer inferred the join to be on column **ID** for you already because both tables had the same column name and data type. Leave that as is.

DFD also chose an **Inner** Join Type, which means only rows that are common to both tables with a matching ID will result. Leave that as is.

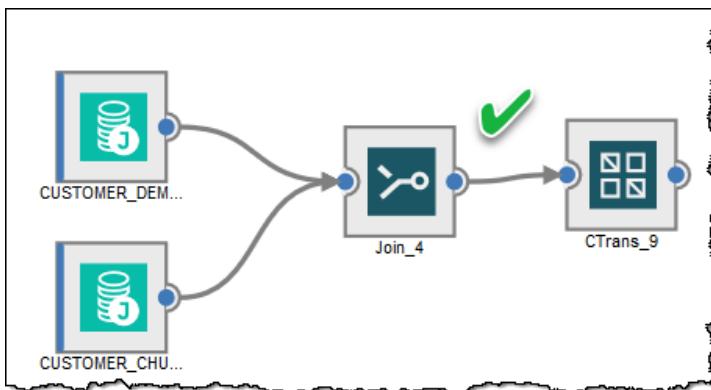
Click **OK** to keep these join options.



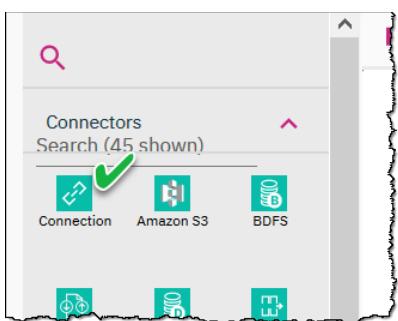
__138. Open the **palette** icon again and choose stage: **Transformer**



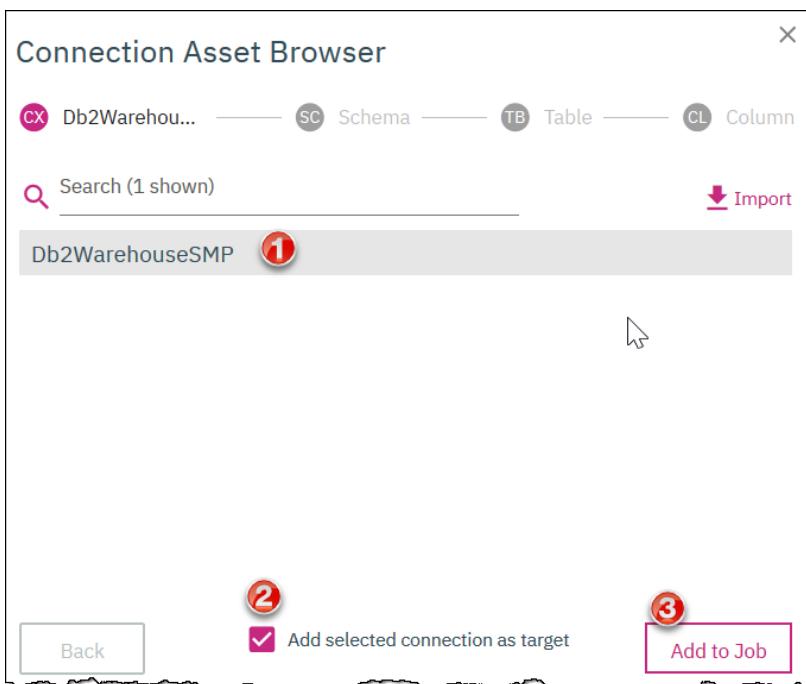
- 139. Drag the **Transformer** icon to the right of the **Join** stage icon. DFD should automatically link it to the Join stage icon. If this doesn't happen, make the link yourself.



- 140. From the **Connectors** palette, choose the **Connection** icon and drag it onto the canvas to the right of the **Transformer** icon

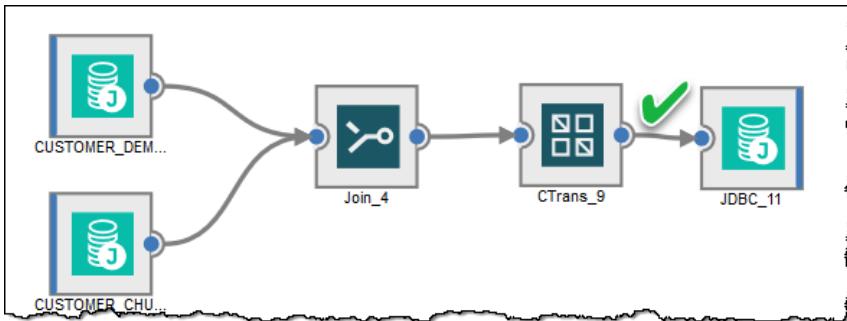


- 141. The **Connection Asset Browser** will pop-up – choose connection **Db2WarehouseSMP** ⇒ **Add selected connection as target** ⇒ **Add to Job**



142. Arrange your canvas icons as shown below. If DFD did not do so for you, connect the **Transformer** icon to the target **JDBC connection** icon.

The icons on the palette should look like this:



Data Steward

The default names given these stages (like **Join_4**, **CTrans_9**) may not exactly match yours. This really doesn't affect anything. You could in fact rename these stages to be more meaningful to you. Leave them as-is for now.

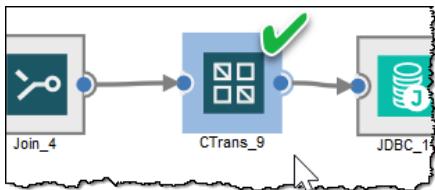
143. Double-click on the **Join** stage and then click on the **Outputs** section which represents the columns coming out of the Join.

Notice that column **CHURNRISK** was from the **CUSTOMER_CHURN** table, whereas all the other columns are from the **CUSTOMER_DEMOGRAPHICS** table. **ID** is the common column the two tables are joined by.

Column	Type	Source Table
ID	SmallInt	Link_5.ID
GENDER	VarChar(1)	Link_5.GENDER
STATUS	VarChar(1)	Link_5.STATUS
CHILDREN	SmallInt	Link_5.CHILDREN
ESTINCOME	Decimal(9,2)	Link_5.ESTINCOME
HOMEOWNER	VarChar(1)	Link_5.HOMEOWNER
AGE	SmallInt	Link_5.AGE
CHURNRISK	SmallInt	Link_5.CHURNRISK

from table customer_churn

_144. Double click on the **Transformer** icon to edit it.



_145. Select the **Inputs** tab to review the columns coming into the **Transformer** stage. Notice the input name is generated as “Link_10”. (Your value may vary – if so, make note of it).

CTrans_9 - Transformer Stage

Properties		Inputs				
Input name		Link_10				
<input type="text"/> Search input columns (8 shown)						
Column name	Key	SQL type	Extended	Length		
ID	false	SMALLINT	true	0		
GENDER	false	CHAR	false	1		
STATUS	false	CHAR	false	1		

_146. Now select the **Outputs** tab, where we will build our transformation of this data. Notice some automatic derivations have been made for you to start with, which are simply the input column name with the Transformer **input name** in front of it. For example, **ID** becomes **Link_10.ID**.

CTrans_9 - Transformer Stage

Properties		Inputs		Outputs				
Output name		Link_12		Jump to Columns				
Constraint				<input type="checkbox"/> Otherwise/Log Abort after rows 0				
<input type="text"/> Search output columns (8 shown)				<input type="checkbox"/> Description <input type="checkbox"/> Load <input type="checkbox"/> Add <input type="checkbox"/>				
Derivation	Column name	Key	SQL type	Extended	Length	Scale	Nullable	
<input type="checkbox"/> Link_10.ID	ID	false	SMALLINT	true	0	0	false	
<input type="checkbox"/> Link_10.GENDER	GENDER	false	CHAR	false	1	0	true	
<input type="checkbox"/> Link_10.STATUS	STATUS	false	CHAR	false	1	0	true	

_147. Click **+Add** to create a new derived column.



- __148. Scroll down to find the new derived column and click in it to edit it.

LINK_8.CHURNRISK	CHURNRISK	rate	VARCHAR	rate
{derivation}	New1	false	CHAR	false

- __149. Fill in the **Derivation** field with the following text.

Note: if your Input name Link_n number is different than this example, use your value and not the one from this example for all three of the highlighted references shown in red.

```
IF Link_8.AGE < 18 THEN "Child" ELSE IF Link_8.AGE < 30 THEN "Young Adult"
ELSE IF Link_8.AGE < 65 THEN "Adult" ELSE "Senior"
```

Derivation Builder - New1 CHAR(0)

Derivation

```
IF Link_8.AGE < 18 THEN "Child" ELSE IF Link_8.AGE < 30 THEN "Young Adult" ELSE IF Link_8.AGE < 65 THEN "Adult" ELSE "Senior"
```

- __150. Double-click **OK** to save the derivation.

- __151. For this derivation output field, click on the temporary column name (**New1**) and change it to **AGE_GROUP**.

LINK_8.AGE	AGE
Link_8.CHURNRISK	CHURNRISK

IF Link_8.AGE < 18 THEN "Child" ELSE IF Link_8.AGE < 30 THEN "Young	AGE_GROUP
---	-----------

- __152. Change the SQL type to **VarChar** and Length to **11**.

SQL type Extent

DECIMAL	
CHAR	false
SMALLINT	true
VARCHAR	false
VarChar	<input checked="" type="checkbox"/>

Length

8
1
0
6
11

__153. The final results should look like this (Note: the link_n.AGE references in the derivation matches the value for Link_n.AGE shown in the output columns)

The screenshot shows the 'Properties' tab of the Transformer stage. The 'Output name' is set to 'Link_10'. Under 'Inputs', there is a constraint section with a toggle for 'Otherwise/Log' and an 'Abort after rows' setting at 0. The 'Outputs' tab is selected, showing a table of output columns. A green arrow points to the 'Link_8.AGE' row. The table has columns: Derivation, Column name, Key, SQL type, Extended, Length, and Scale. The 'Link_8.AGE' row has 'AGE' in the Column name column and 'SMALLINT' in the SQL type column. The 'Link_8.CHURNRISK' row has 'CHURNRISK' in the Column name column and 'VARCHAR' in the SQL type column. The 'IF Link_8.AGE < 18 THEN "Child" ELSE IF Link_8.AGE < 30 THEN "Young" ELSE "Adult"' row has 'AGE_GROUP' in the Column name column and 'VARCHAR' in the SQL type column. The 'Length' column shows 6 and 11, and the 'Scale' column shows 0 and 0.

Derivation	Column name	Key	SQL type	Extended	Length	Scale
Link_8.ESTINCOME	ESTINCOME	false	DECIMAL		8	2
Link_8.HOMEOWNER	HOMEOWNER	false	CHAR	false	1	0
Link_8.AGE	AGE	false	SMALLINT	true	0	0
Link_8.CHURNRISK	CHURNRISK	false	VARCHAR	false	6	0
IF Link_8.AGE < 18 THEN "Child" ELSE IF Link_8.AGE < 30 THEN "Young" ELSE "Adult"	AGE_GROUP	false	VARCHAR	false	11	0

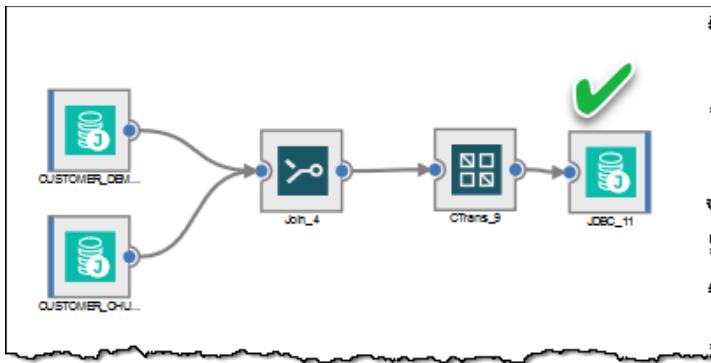
__154. Click **OK** to save your work in the **Transformer** stage.



__155. Save this job again to preserve the work you have done so far.



__156. Now click on the output (target) connection stage icon.



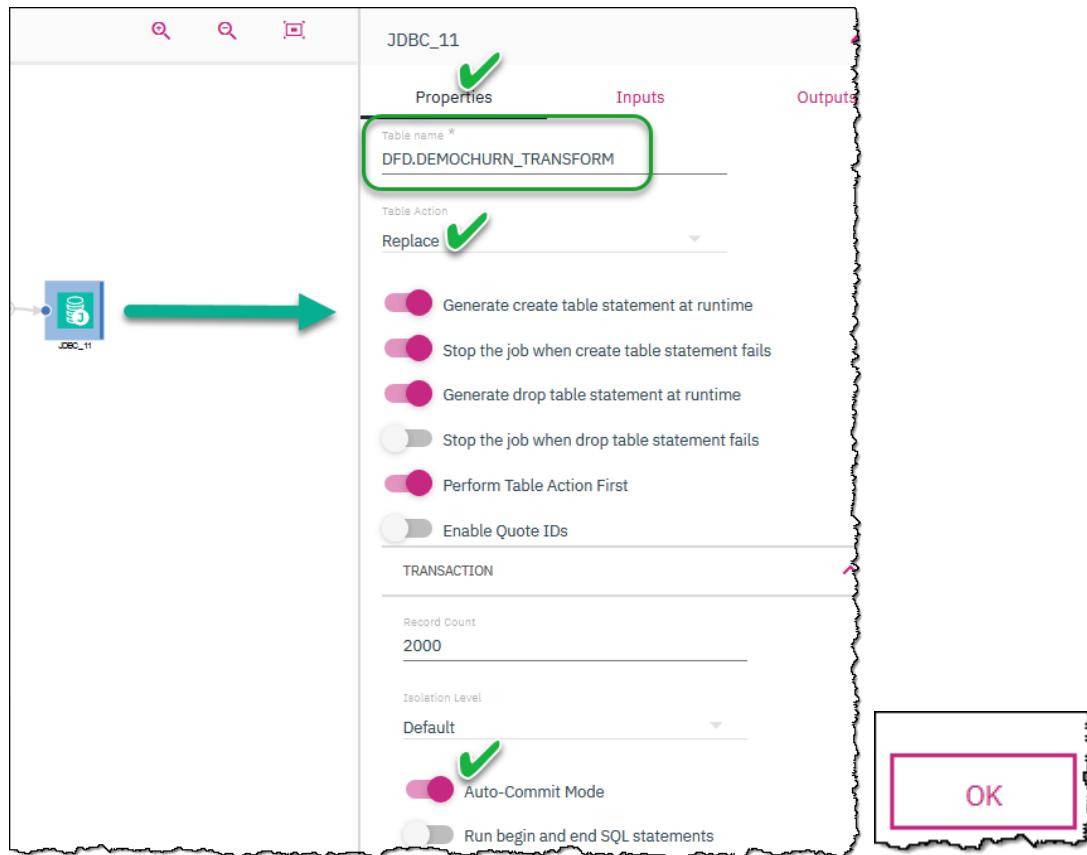
_157. Click on **Properties** tab and scroll down to find Table name.

Fill Table name in as **DFD.DEMOCHURN_TRANSFORM**.

In Table Action, choose **Replace** (Leave the defaults for this option as-is)

Slide on the button: **Auto-Commit Mode**

Click **OK** to save.



_158. Save this job again.



_159. This enables the **Compile** option (the wrench icon) Click on it to Compile your job.

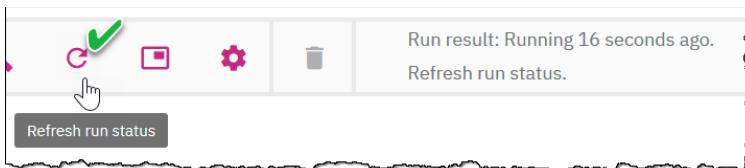


__160. After it compiles successfully, click on the Run icon (the running man), then Run

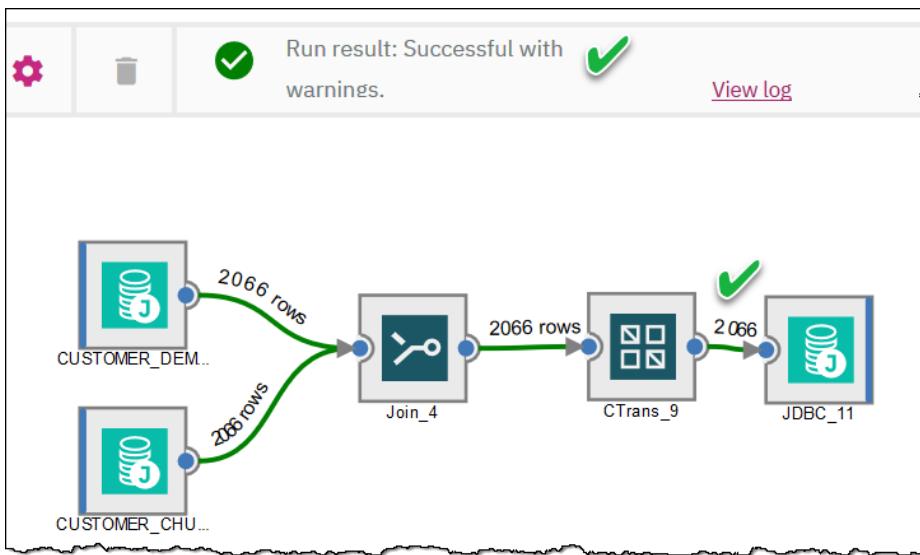


Run

__161. Let the job run for 1 minute, then click on the Refresh run status icon

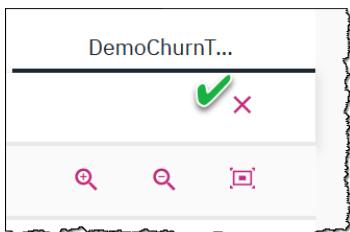


__162. You should now see the job successfully finish with a record of the rows that flowed through the job and written to the output table in the target connection stage.



 Data Steward	<p>If the job did not complete successfully, check for these common mistakes:</p> <ul style="list-style-type: none"> • Make sure the input and output tables in the connections use the schema DFD (entered before the table name) • Make sure the connections between the stages are made as shown in the above lab steps • Make sure your Transformer stage derived column formula is correct.
 Data Steward	<p>Reviewing the View Log should produce only one warning and that is the Db2 drop table statement for the output table created from the run. This is normal because the output table did not exist on the first run. On subsequent runs this warning would not be there.</p>

__163. Close the job window to end your edit session.



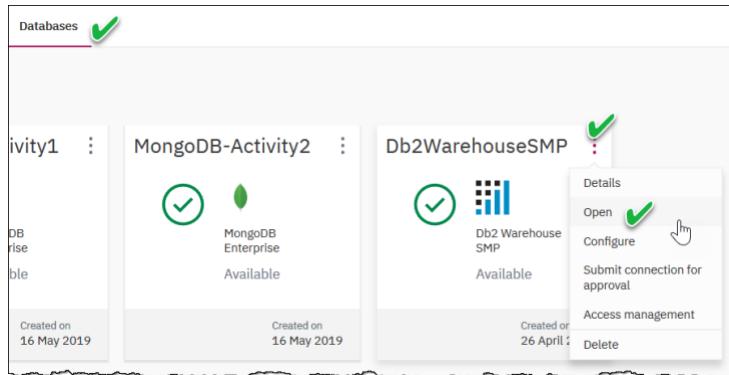
__164. You will see the job `DemoChurnTransform` you just created in project `dstage1`.

Name	Modified on	Category
CustomerJoin	2019-05-06 03:43:26	\Jobs
DemoChurnTransform	2019-05-21 16:55:14	\Jobs

__165. Let's make a check on our data by reviewing the Db2 Warehouse again:

Navigation Menu \Rightarrow Collect \Rightarrow My data \Rightarrow Databases

Db2WarehouseSMP \Rightarrow ellipsis \Rightarrow Open



__166. Then Menu \Rightarrow Tables



__167. Schema DFD \Rightarrow DEMOCHURN_TRANSFORM

The screenshot shows a schema browser interface. On the left, under 'Schemas', the 'DFD' schema is selected. On the right, under 'Tables', the 'DEMOCHURN_TRANSFORM' table is selected, indicated by a green checkmark. Other tables listed include 'NAME', 'CUSTOMER_CHURN', 'CUSTOMER_DEMOGRAPHICS', and 'ICPD'.

__168. Then View Data

[View Data](#)

__169. Browsing through this transformed output table shows our derived column AGE_GROUP

The screenshot shows a data preview table for the 'DEMOCHURN_TRANSFORM' schema. The table has columns: AGE_GROUP, ID, GENDER, STATUS, CHILDREN, ESTINCOME, HOMEOWNER, and AGE. The 'AGE_GROUP' column is highlighted with a green border. The data rows show various demographic information for different individuals, including their age group (e.g., Adult), gender, status, number of children, estimated income, homeownership status, and age.

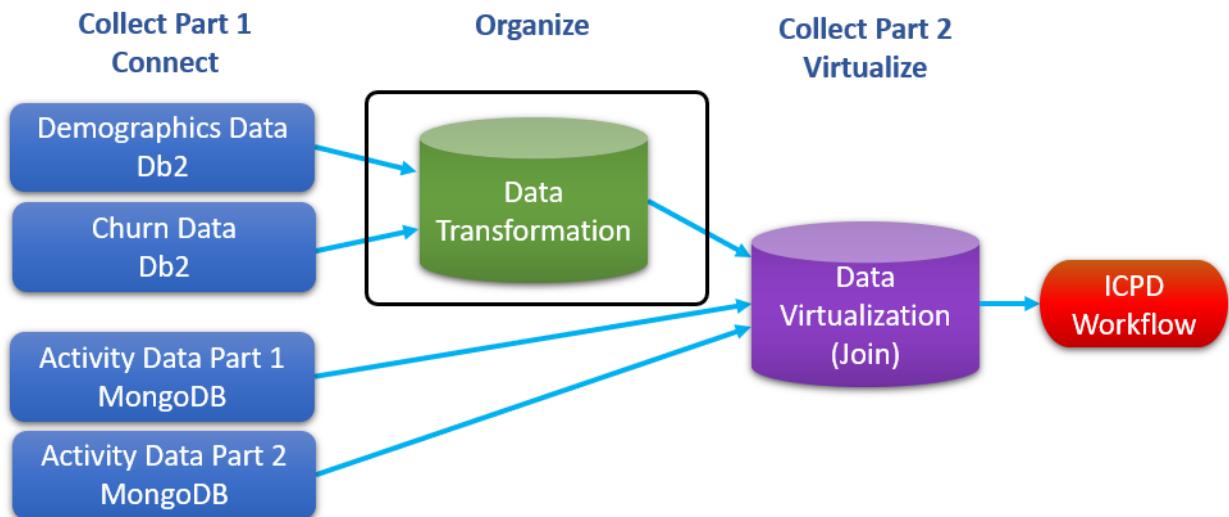
AGE_GROUP	ID	GENDER	STATUS	CHILDREN	ESTINCOME	HOMEOWNER	AGE
Adult	409	F	M	2	82644.20	Y	33
Adult	296	F	S	1	13576.50	N	39
Adult	295	M	S	2	89459.90	N	53
Adult	294	F	M	2	5237.63	N	49
Adult	27	F	S	1	16326.70	N	51
Adult	292	F	M	2	28220.80	N	39

4.8 Lab conclusion

We have seen the value in creating a [Data Dictionary](#) by creating a [glossary of categories](#) and [terms](#) to make data searchable so that data scientist, data engineers and business analysts can shop for data. The CPD platform reduces the time-consuming data organization task, making it easier for the consumers of the data.

This lab has shown you how to automatically [discover](#) data sources, automatically classify those sources with business classifications using CPD machine learning methods and automatically assign business terms to those data sources.

It showed you how to [shop for specific data](#), and join and move that data to an analytics warehouse in a [Data Transformation](#) step for further use by the data consumers in the CPD workflow.



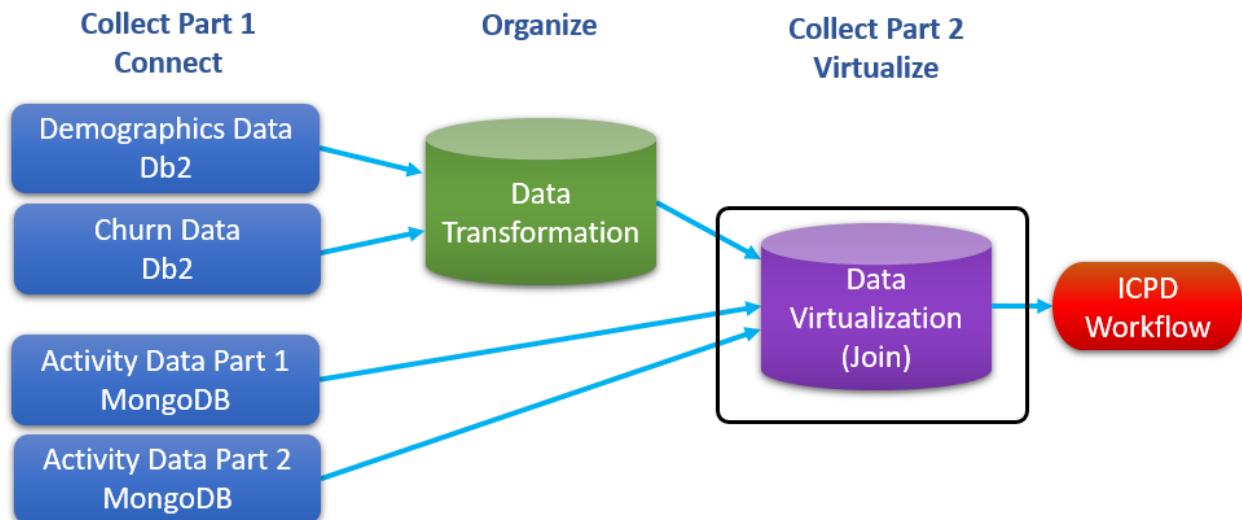
The steps covered here could normally take months, and sometimes years, to complete using traditional manual methods. CPD automates these things so that you can accelerate the time to value of your analytics projects.

**** End of Lab 04: Organize**

Lab 05 Collect Part 2 - Virtualize

5.1 Lab overview

In this lab, you will learn about [Data Virtualization](#) to complete the [Collect](#) tasks by creating a virtualized view of the transformed Db2 Demographics and Churn data, joined with the MongoDB Activity data.



5.2 Persona represented in this lab

The [Data Engineer](#) persona will be taking over again to perform the various [Collect](#) tasks shown in this lab.

Persona (Role)	Capabilities
 Data Engineer	Data Engineers build and optimize the systems to allow data scientists and business analysts to perform their work. The Data Engineer ensures that any data is properly received, transformed, stored, and made accessible to other users.

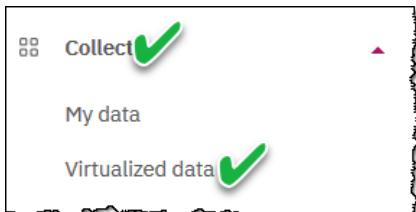
Note: Data Virtualization capabilities could be granted to the Data Steward as well, but in this workshop scenario we are assigning this task to the Data Engineer.

5.3 Data Virtualization data sources

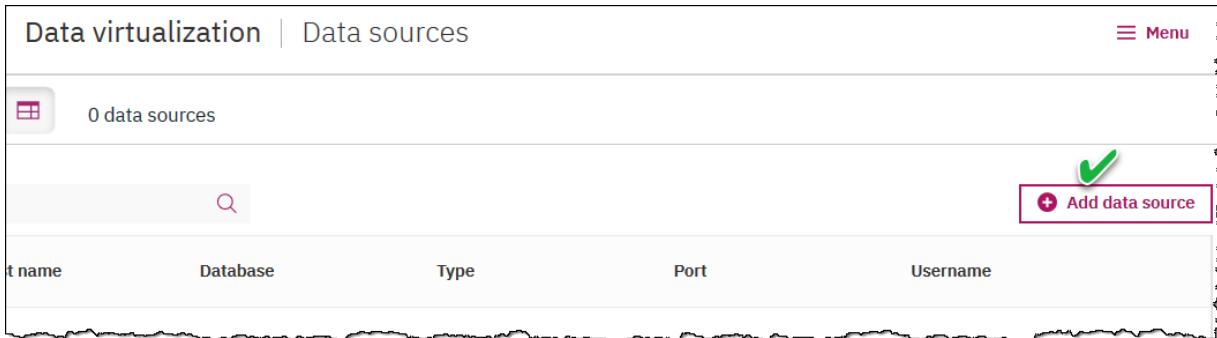
Let's explore the Data Virtualization process by adding data sources to virtualize.

_170. Start at the [Navigation Menu](#)

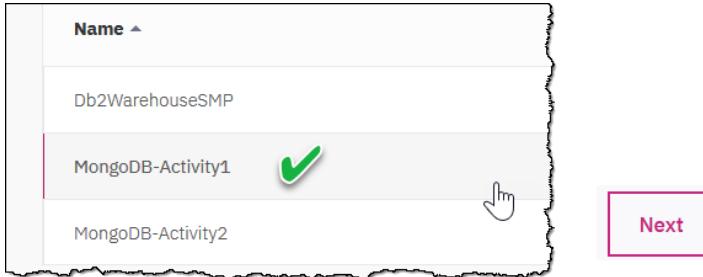
Click [Collect](#) ⇒ [Virtualized data](#)



_171. From the Data sources screen, click + [Add data source](#)



_172. Select (click on) [MongoDB-Activity1](#) ⇒ [Next](#)

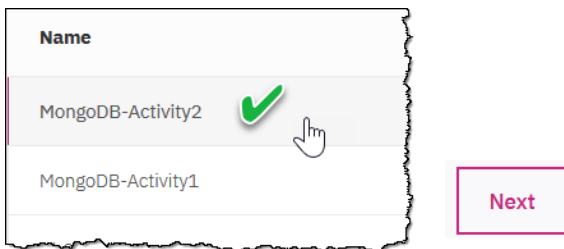


 Data Engineer	<p>If any of the Add data source activities fails in this exercise, go back to the Navigation menu ⇒ Home. Then enter the Virtualized data and try again.</p>
---	---

_173. For a second time, click + [Add data source](#)



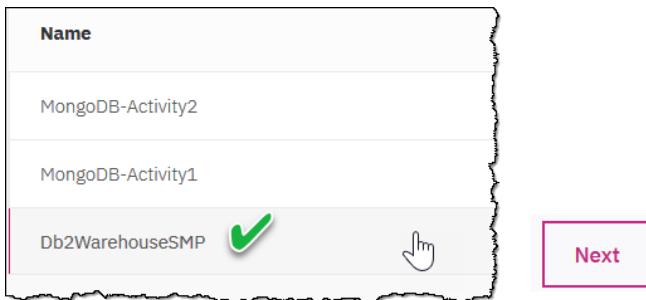
_174. Select (click) MongoDB-Activity2 \Rightarrow Next



_175. For a third time, click + Add data source



_176. Select (click) Db2WarehouseSMP \Rightarrow Next



_177. The Data Virtualization | Data sources screen should have three connections, as shown here. They are two Mongo and one Db2 source. (Hint: make sure the Mongo Host names are different)

Data virtualization Data sources				
3 data sources				
Host name	Database	Type	Port	Username
192.168.142.21	BLUDB	Db2 Family	32267	icpd
10.1.104.76	mongodb	Mongo DB	27017	admin
10.1.104.79	mongodb	Mongo DB	27017	admin



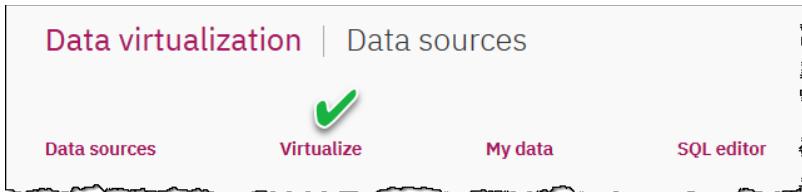
Even though you used previously created connections to create these sources, the ones shown here are now specific to Data Virtualization processing.

5.4 Virtualize the MongoDB data with the Db2 data

_178. Click **Menu** to bring you to the Data Virtualization options.



_179. Click **Virtualize**



_180. In the search box, type in **trader**

Check the box for **Mongo DB (2)** and click the search icon

Filters Available tables Automatically group tables

Databases

- MongoDB
- Mongo DB (2) **(highlighted with a green checkmark)**
- IBM BigSQL (0)
- IBM

Available tables

Table	Schemas	Databases
TRADERINFO	MONGODB	mongodb
TRADERINFO	MONGODB	mongodb

181. Check the box **Automatically group tables**

Notice that tables with the exact same name (regardless of the database and schema they reside in) will be grouped (or folded) together. This essentially makes them one table.

Available tables Automatically group tables

1 tables

Table	Schemas	Databases	Grouped tables <small>i</small>
TRADERINFO	MONGODB,MONGODB	mongodb,mongodb	2


Data Engineer

You could have grouped many more databases than the two in this workshop.

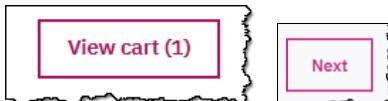
182. Now check the table **TRADERINFO** and click **Add to cart**

Available tables Automatically group tables

1 tables

Table	Schemas	Databases	Grouped tables <small>i</small>
TRADERINFO <input checked="" type="checkbox"/>	MONGODB,MONGODB	mongodb,mongodb	2

View cart (0) 2 ✓ Add to cart

183. Click **View cart** then **Next**184. Under **Name**, fill in the name for this virtualized table: **ALL_ACTIVITY_VIRTUALIZED**

Click **Next**

Virtualize: Review

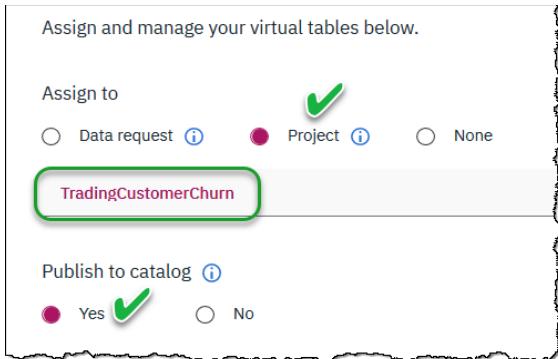
Review your selected objects and confirm table and schema names.

Name	Schema	Source schema
ALL_ACTIVITY_VIRTUALIZED	USER999	MONGODB,MONGODB

Next

_185. Click button **Assign to** \Rightarrow **Project**, then choose project: **TradingCustomerChurn**

Click button **Publish to catalog** \Rightarrow **Yes**



Click **Virtualize**



_186. The virtualized table of the two MongoDB sources was created.

Click **Virtualize more data**

Virtual tables created

1 of 1 tables successfully virtualized.

Table	Schema	Status
ALL_ACTIVITY_VIRTUALIZED	USER999	success

Assigned to project

TradingCustomerChurn

[View my data](#) [Virtualize more data](#) [Go to project](#)

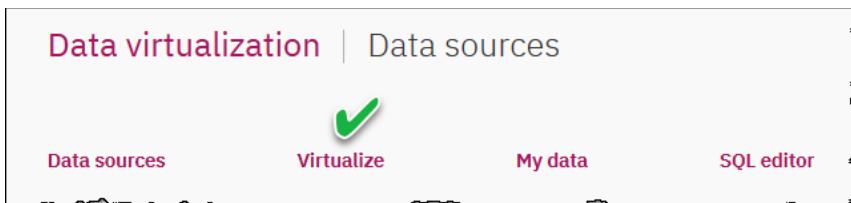
5.5 Virtualize the Db2 data

Now you will virtualize the Db2 table `DEMOCHURN_TRANSFORM` that you created in the Organize lab to prepare it for a join with the `ALL_ACTIVITIES_VIRTUALIZED` table.

- _187. At the top right of the screen click on the hamburger icon **Menu**.



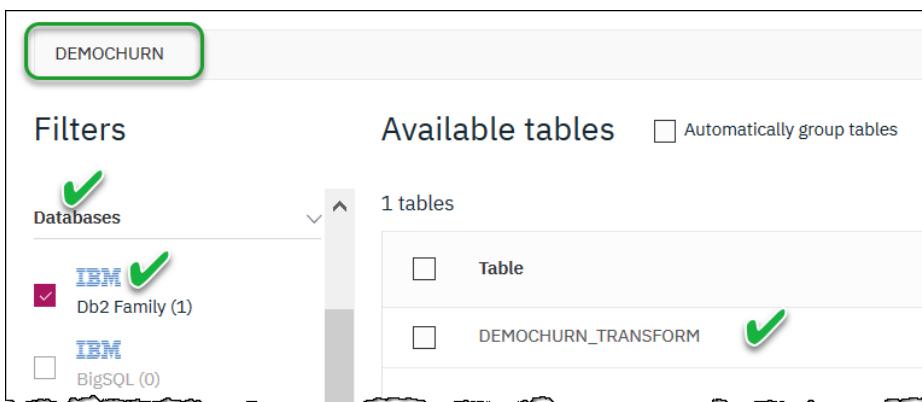
- _188. Click **Virtualize**



- _189. Under **Databases**, check box **Db2 Family** (Make sure Mongo DB is not checked.)

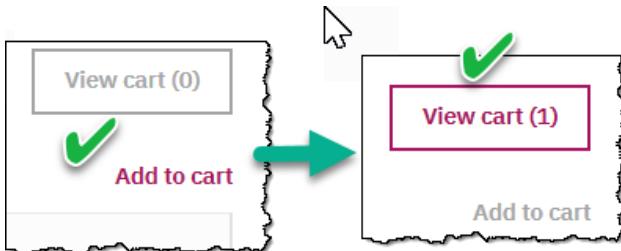
Fill in **Search** bar with DEMOCHURN then hit **Enter**

Results should show table `DEMOCHURN_TRANSFORM`



- _190. Check table `DEMOCHURN_TRANSFORM` then Add to cart

Click on **View cart (1)**



_191. In Virtualize Review screen, change the Name to DEMOCHURN_VIRTUALIZED

Click [Next](#)

Virtualize: Review

Review your selected objects and confirm table and schema names.

Name	Schema	Source schema
DEMOCHURN_VIRTUALIZED	USER999	DFD

Next

_192. Click on [Assign to](#) button ⇒ [Project](#), then choose project: **TradingCustomerChurn**

Click on [Publish to catalog](#) button ⇒ **Yes**

Assign and manage your virtual tables below.

Assign to

Data request [i](#) Project [i](#) None

TradingCustomerChurn

Publish to catalog [i](#)

Yes No

Click [Virtualize](#)



_193. Click [View my data](#)

Virtual tables created

1 of 1 tables successfully virtualized.

Table	Schema	Status
DEMOCHURN_VIRTUALIZED	USER999	success

Assigned to project
TradingCustomerChurn

[View my data](#) [Virtualize more data](#) **Go to project**

5.6 Join the two virtualized tables

Now we will join the two virtualized tables `ALL_ACTIVITY_VIRTUALIZED` and `DEMOCHURN_VIRTUALIZED`

- _194. If you did not choose `View my data` from the previous screen, from the Virtualization screen you can select `Menu` ⇒ `My data`



- _195. Check the two virtualized tables you just created in the previous exercises

Click on `Join view`

A screenshot of the Data virtualization interface under the 'My data' section. The title is 'Data virtualization | My data'. Below it is a search bar and buttons for 'Assign', 'Join view' (which is highlighted with a red box), and 'Add table'. A red circle with the number '3' is on the top right. The main area lists two tables: 'DEMOCHURN_VIRTUALIZED' (marked with a red circle '1') and 'ALL_ACTIVITY_VIRTUALIZED' (marked with a red circle '2'). Both have 'USER999' as the schema and were created on '22 May 2019 13:42:12' and '17 May 2019 03:49:22' respectively.

- _196. Click and hold on `ID` in the first table and drag it to `ID` in the second to connect them.

Note: the order of columns on your screen may appear differently than this example.

A screenshot of the Data virtualization interface showing the 'Join view' operation. On the left, 'Table 1: DEMOCHURN_VIRTUALIZED' is shown with columns: Column Name (checkbox) and Data Type. Rows include AGE (SMALLINT), AGE_GROUP (VARCHAR), CHILDREN (SMALLINT), CHURNRISK (VARCHAR), ESTIMATED_INCOME (DECIMAL), GENDER (CHAR), HOME_OWNER (CHAR), ID (SMALLINT), and STATUS (CHAR). On the right, 'Table 2: ALL_ACTIVITY_VIRTUALIZED' is shown with columns: Column Name (checkbox) and Data Type. Rows include DAYSSINCELASTLOGIN (INTEGER), DAYSSINCELASTTRADE (INTEGER), ID (INTEGER), LARGESTSINGLETRANSACTION (DOUBLE), NETREALIZEDGAINS_YTD (DOUBLE), NETREALIZEDLOSSES_YTD (DOUBLE), PERCENTCHANGECALCULATION (DOUBLE), SMALLESTSINGLETRANSACTION (DOUBLE), and TOTALDOLLARVALUETRADED (DOUBLE). A blue line with a hand icon labeled 'drag' connects the 'ID' column in Table 1 to the 'ID' column in Table 2. A red circle with 'click' and a hand icon is placed over the 'ID' column in Table 1.

__197. This should populate the [Join keys](#) as shown below.

Join keys	
DEMOCHURN_VIRTUALIZED	ALL_ACTIVITY_VIRTUALIZED
SMA_ID	INT_ID

__198. Scroll to find column [_ID](#) in the virtualized table [ALL_ACTIVITY_VIRTUALIZED](#)

Uncheck it. This column is a generated column by Mongo and you will not be needing it for analytics processing.

<input type="checkbox"/>	Column Name	Data Type
<input checked="" type="checkbox"/>	DAYSSINCELASTTRADE	INTEGER
<input checked="" type="checkbox"/>	ID	INTEGER
<input checked="" type="checkbox"/>	LARGESTSINGLETRANSACTION	DOUBLE
<input checked="" type="checkbox"/>	NETREALIZEDGAINS_YTD	DOUBLE
<input checked="" type="checkbox"/>	NETREALIZEDLOSSES_YTD	DOUBLE
<input checked="" type="checkbox"/>	PERCENTCHANGECALCULATION	DOUBLE
<input checked="" type="checkbox"/>	SMALLESTSINGLETRANSACTION	DOUBLE
<input checked="" type="checkbox"/>	TOTALDOLLARVALUETRADED	VARCHAR
<input checked="" type="checkbox"/>	TOTALUNITSTRADED	VARCHAR
<input type="checkbox"/>	_ID	VARCHAR

_ID is a
generated
column from
Mongo - it is
not needed
for analytical
processing

__199. Click [Preview](#)



The [Preview](#) of the [Join](#) should look similar to this

AGE	AGE_GROUP	CHILDREN	CHURNRISK	DAYSSINCELASTLOGIN	DAYSSINCELASTTRADE	ESTINCOM
22	Young Adult	0	Low	1	6	37000.00
49	Adult	2	Medium	1	14	48322.50
50	Adult	2	Medium	2	11	81913.80

__200. Scroll to review the columns in the virtualized table you will be creating. [_ID](#) should not be there.

New Join Preview				
LCULATION	SMALLESTINGLETRANSACTION	STATUS	TOTALDOLLARVALUETRADED	TOTALUNITSTRADED
1148.115		S	22962.3	75
167.26675		M	6690.67	43

If your join preview does not produce data (it is blank) one or more of your virtualized tables may not have been created correctly. To preview each virtualized table, go to:

[Menu](#) \Rightarrow [My data](#)

Preview each virtualized table to make sure they show data.

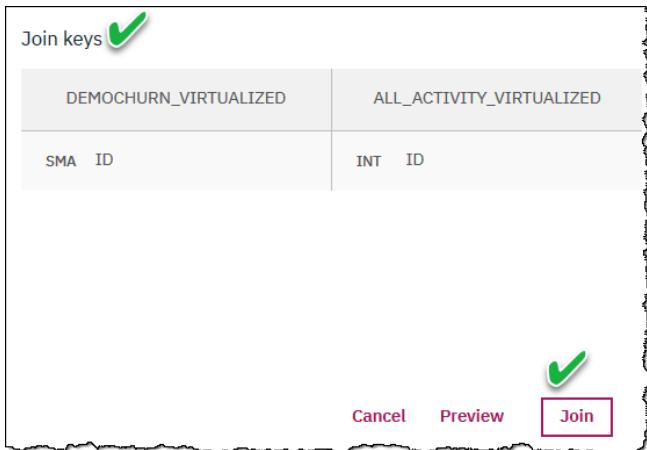


Data
Engineer

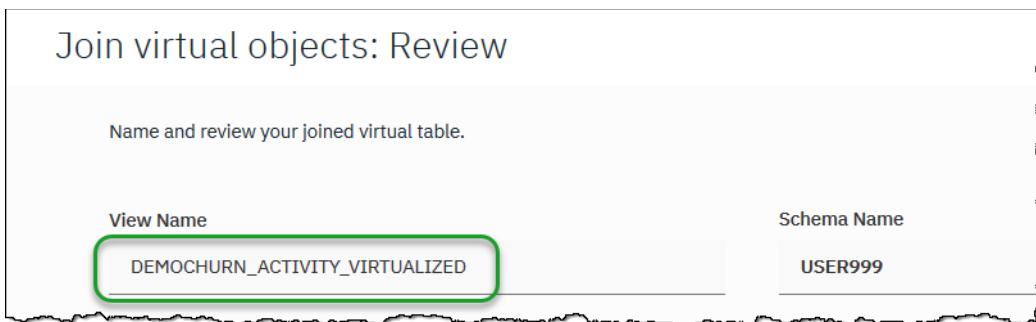
New view
Manage access
View details
Preview ✓
Submit to catalog
Remove

Then recreate the virtualized table that is not showing data, rejoin them, and preview again.

_201. If the preview looks good, click **Join**



_202. Fill in View Name **DEMOCHURN_ACTIVITY_VIRTUALIZED**

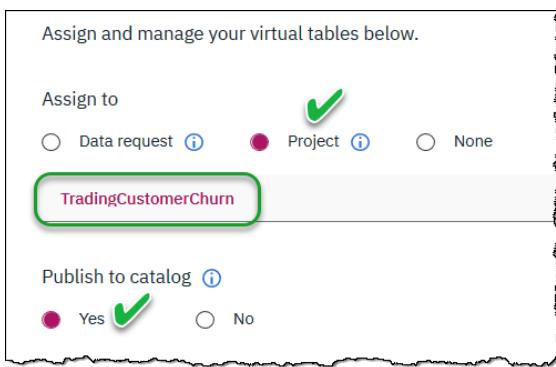


_203. Click **Next**



_204. Click button **Assign to → Project**, then choose project: **TradingCustomerChurn**

Click button **Publish to catalog button → Yes**



Click **Create view**



_205. Review the message and then click [Go to project](#)

Join view created

View DEMOCHURN_ACTIVITY_VIRTUALIZED **Schema** USER999

Assigned to project TradingCustomerChurn **Published to catalog** Succeeded

[Virtualize more data](#) [View my data](#) [Go to project](#) ✓


Data Engineer

Notice that CPD refers to joins of individual virtualized tables as “virtualized views”.

_206. [Preview](#) your new virtualized view in your project.

You will be using this data in the next lab to produce a Machine Learning model

TradingCustomerChurn

Created by admin on 22 May 2019, 3:57

[Assets](#) 13 [Data Sources](#) 2 [Jobs](#) 0 [Environments](#) 1 [Collaborators](#) 1

Recent

Data sets 9

Name	Type	Size	Data Source	Last Modified
USER999.DEMOCHURN_ACTIVITY_VIRTUALIZED	TABLE	—	VT1558562413194	23 May 2019, 1:32 PM
USER999.DEMOCHURN_TRANSFORM_VIRTUALIZED	TABLE	—	VT1558562413194	23 May 2019
USER999.ALL_ACTIVITY_VIRTUALIZED	TABLE	—	VT1558562413194	22 May 2019

+ Add Data Set

Preview

Edit settings

__207. Remember this virtualized view represents:

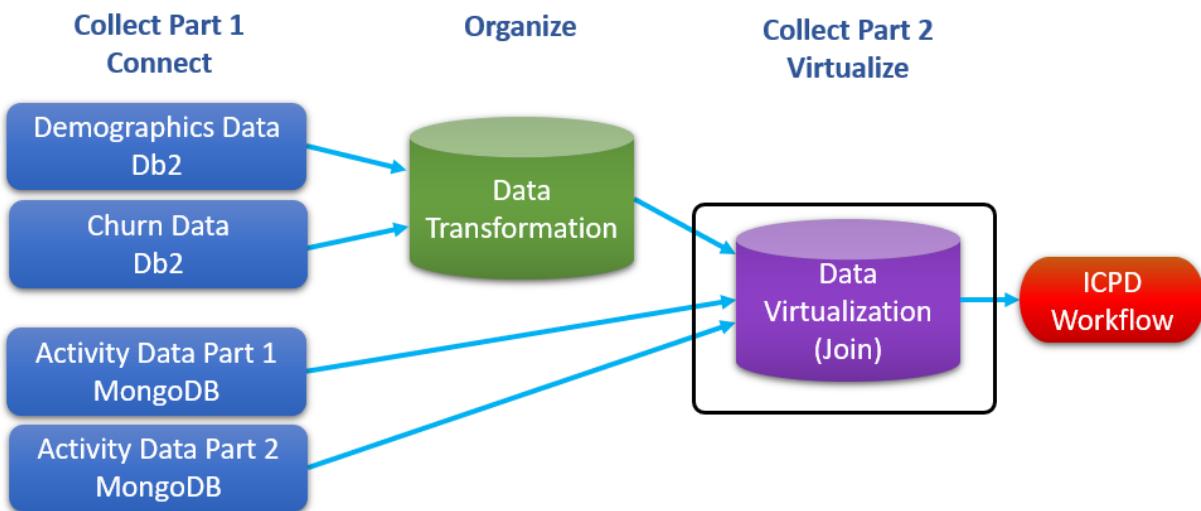
- 1) All the Db2 demographics data (with one column transformed)
- 2) All the Db2 Churn data
- 3) All the MongoDB Activity data (folded together from two databases)

Preview - USER999.DEMOCHURN_ACTIVITY_VIRTUALIZED

AGE	AGE_GROUP	CHILDREN	CHURNRISK	ESTIMATED_INCOME	GENDER	HOMEOWNER	ID	STATUS	DAYSSINCELASTLOGI
24	Adult	1	Low	38000.00	F	N	0	S	3
65	Adult	1	Medium	75004.50	M	N	6	M	4
60	Adult	0	High	19749.30	M	N	7	M	3

5.7 Lab conclusion

In this lab, you learned about [Data Virtualization](#) to complete the [Collect](#) tasks by creating a virtualized table of the transformed Db2 Demographics and Churn data, joined with the MongoDB Activity data.



** End of Lab 05: Collect Part 2 - Virtualize

Lab 06 Analyze Part 1 - Dashboards

6.1 Lab overview

Analyze is the third phase in the CPD platform and workflow. This is where business analysts and data scientists join forces to gain insights from their organization's data.

When embarking on machine learning projects, many organizations immediately engage these personas in hopes that they can quickly gain insight into their data, only to realize that the data they need is not available in the format required. This is where CPD helps because the Collect and Organize steps we have seen in the previous labs easily prepares the data for analytic processing.

6.2 Persona represented in this lab

The Business Analyst persona is the likely to perform the exercises in this Analyze part 1 lab, and that is to create dashboards to make sense of the problems the organization is facing.

Persona (Role)	Capabilities
 Business Analyst	Business Analysts deliver value by taking data, using it to answer questions, and communicating the results to help make better business decisions.

6.3 Dashboards to help identify the focus area

In the Executive Demo lab, you were presented with a Stock Trade Opening Bell business analysis that painted the picture of how the business was doing. The dashboard provided showed a flat shares-sold, and a declining number of daily traders. A Boatswain Trading business analyst provided this information to his executives.



In the previous Collect Part 1 lab, we used Activity, Demographics and Churn data together. In this fictitious scenario, the Churn data shows **Low**, **Medium** and **High** risk of the customer potentially leaving Boatswain Trading.

The output is a dataset called **MERGED_DEMOGRAPHICS_TRADING_CUSTOMER** that business analyst used to build a dashboard to better understand the details of the problem.

Let us see how this looks.

6.3.1 Business dashboard – analyze customer demographics

- _208. From the **Navigation Menu** \Rightarrow **Projects** and select project **TradingCustomerChurn**.

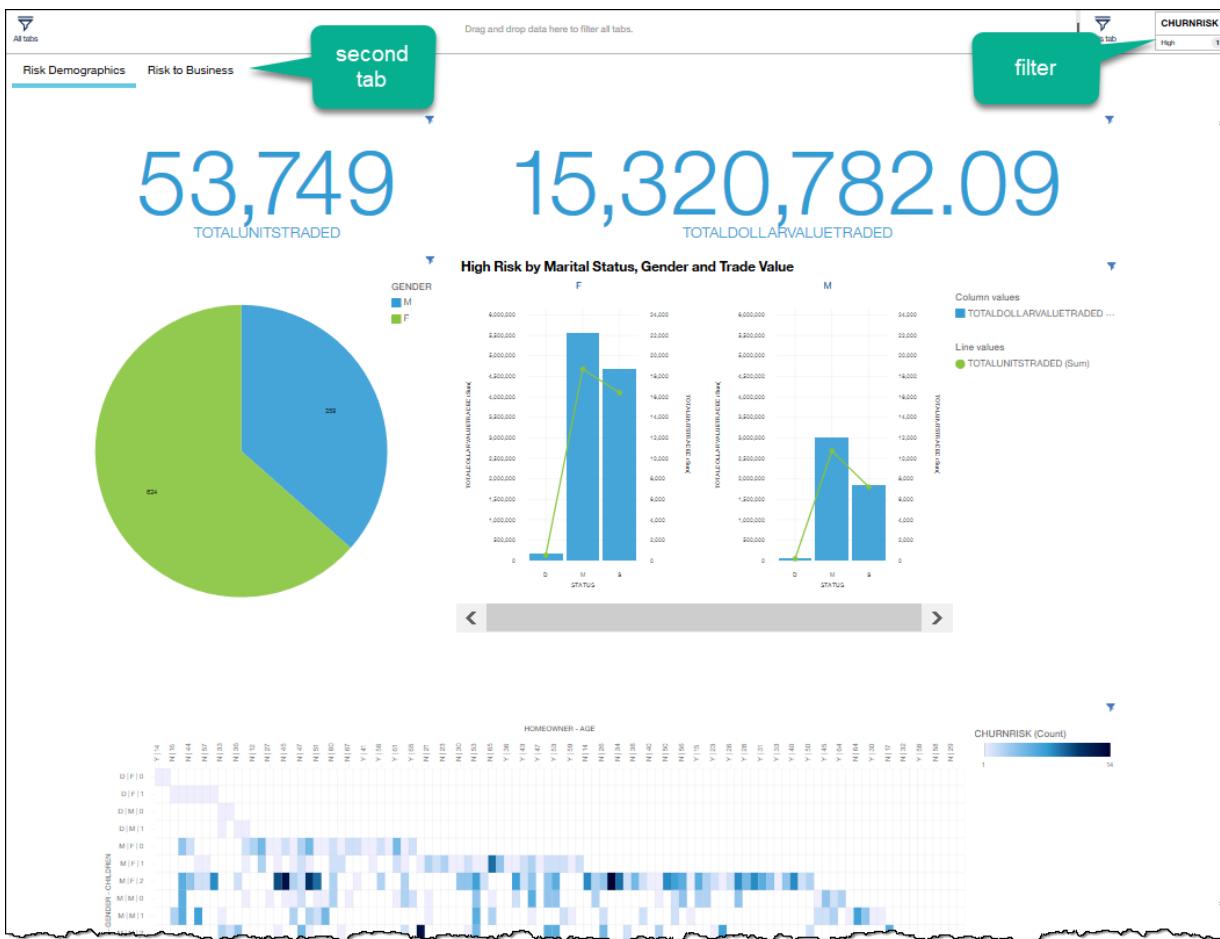
A screenshot of a web-based project management interface. At the top left is a back arrow icon and the word "Projects". Below is a search bar with a magnifying glass icon. A table lists projects with columns "Name" and "Project Type". The row for "TradingCustomerChurn" has a green checkmark icon to its left and "Analytics" listed under "Project Type".

- _209. Click **Analytics Dashboards** \Rightarrow **02-Stock-Trader-Demographic-Discovery**

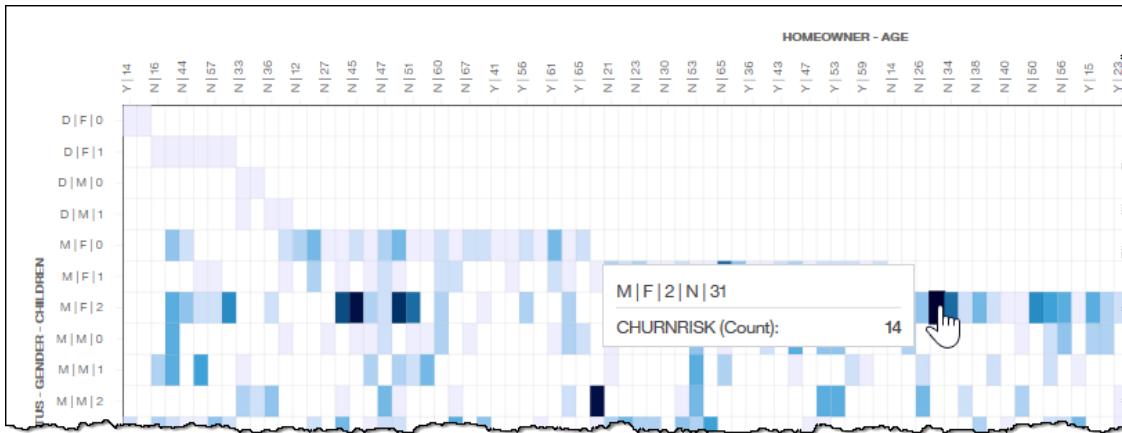
A screenshot of a web-based analytics dashboard interface. On the left is a sidebar with "Recent" items and categories like "Data sets" (4), "Notebooks" (1), "Scripts" (0), "Models" (0), "Model groups" (0), and "Analytics dashboards" (3). The "Analytics dashboards" item is highlighted with a green checkmark icon. To the right is a main panel titled "Analytics dashboards" with a count of 3. It lists three dashboards: "01-Stock-Trade-Opening-Bell", "02-Stock-Trader-Demographic-Discovery" (which has a green checkmark icon to its right), and "03-Stock-Trader-Closing-Bell".

 Business Analyst	<p>The Analytics dashboards you will be using in this exercise use the built-in Cognos dashboards of CPD.</p> <p>You can install and enable the Cognos Analytics add-on for more power and capability, with features such as:</p> <ul style="list-style-type: none"> • Automated data preparation, modeling and modules • Automated creation of visualizations and dashboards • Reporting and stories • Data exploration with a natural language interface • Dynamic query
---	--

—210. This is the dashboard that the Boatswain Trading business analyst created to break down the demographics of their customer base, based on the likeliness of churn.



- _211. Note that this dashboard is interactive. For example, if you click on a dark square in the heatmap (the darker the square, the more likely to churn) you can see in the example below that this data point represents a married female with 2 children, not a homeowner, and is 31 years of age. The rest of the visualizations on this tab will change based on this selection as well, which provides even more insight into this data point.



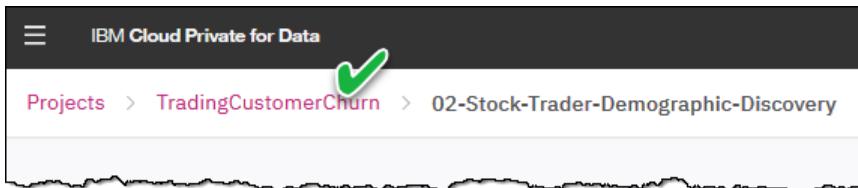
Dashboards like this can paint a high-level picture of the customer demographic Boatswain Trading needs to target in order to reverse their declining revenue trend.

Note too that you can change the CHURNRISK filter using **Low**, **Medium** and **High** to help better understand the data.

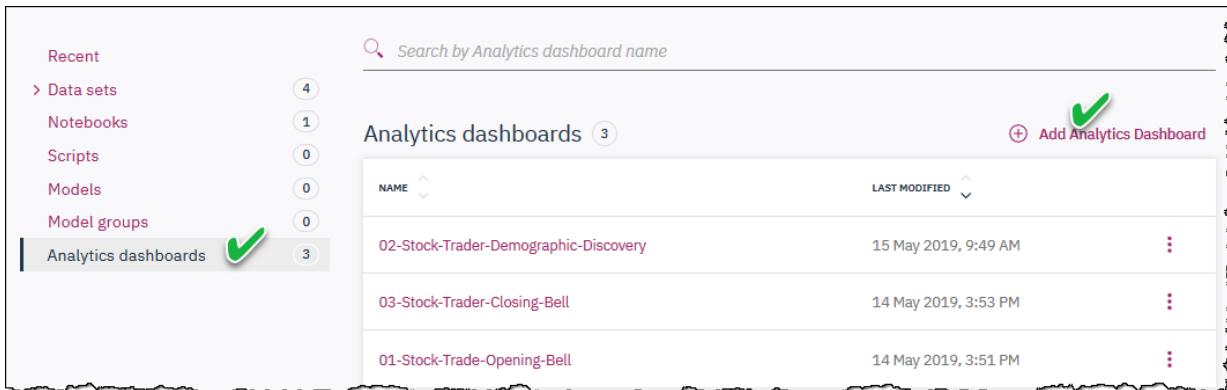
6.3.2 Building the dashboard

Let us build this dashboard from the beginning to understand how this works in CPD.

- _212. In the navigation bar on the top right, click on the project: **TradingCustomerChurn**.



- _213. Click **Analytics dashboards** ⇒ **Add Analytics Dashboard**



_214. Name this new dashboard: [My-Stock-Trader-Demographic-Discovery](#) and click [Create](#).

Create Dashboard

Blank From File

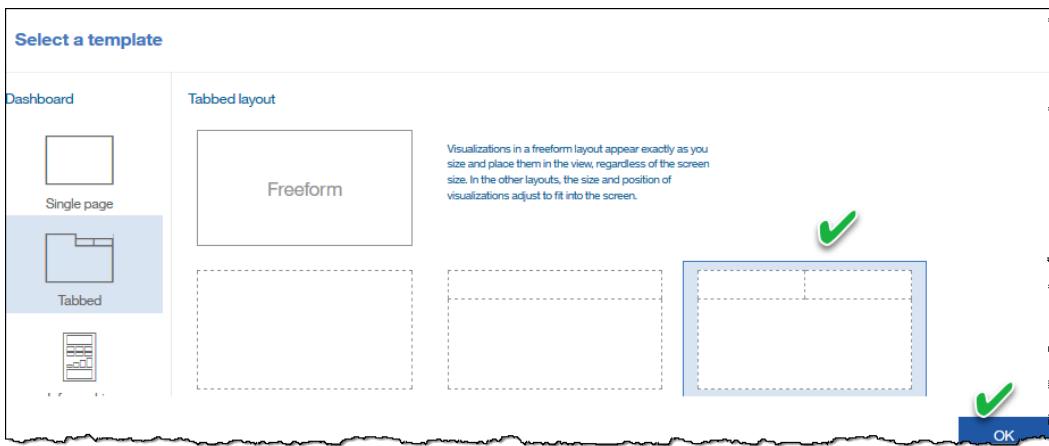
Name*
My-Stock-Trader-Demographic-Discovery

This name is valid 13

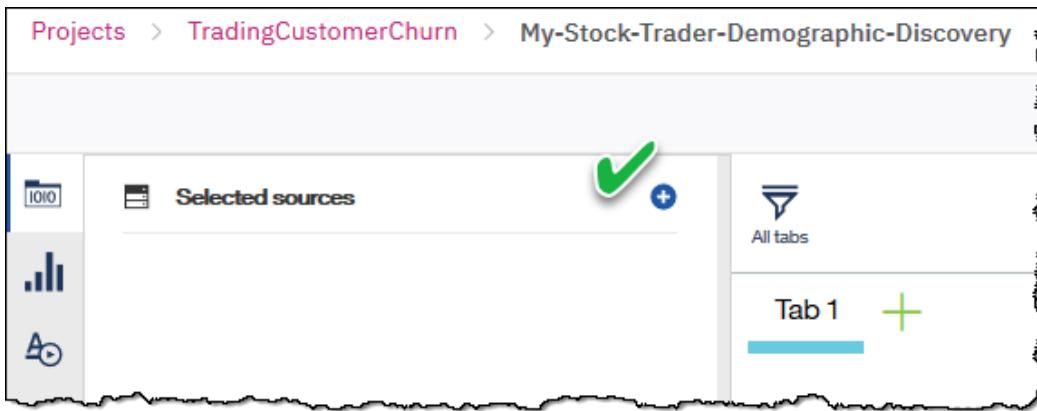
Description
Type your description here

Cancel Create

_215. Select the template shown below and click [OK](#).

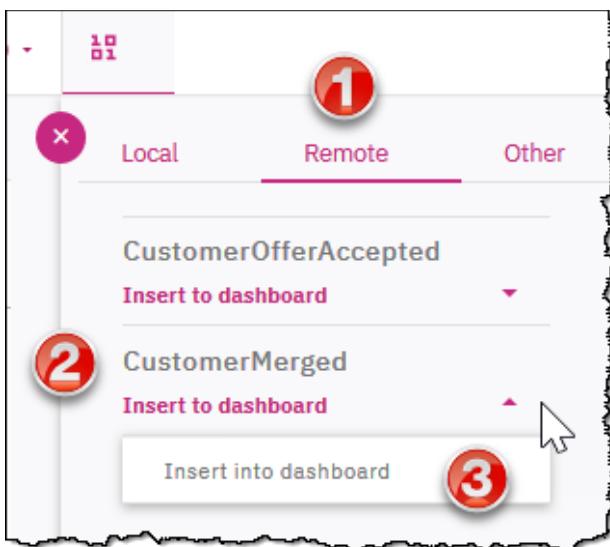


_216. Click the + next to **Selected sources**



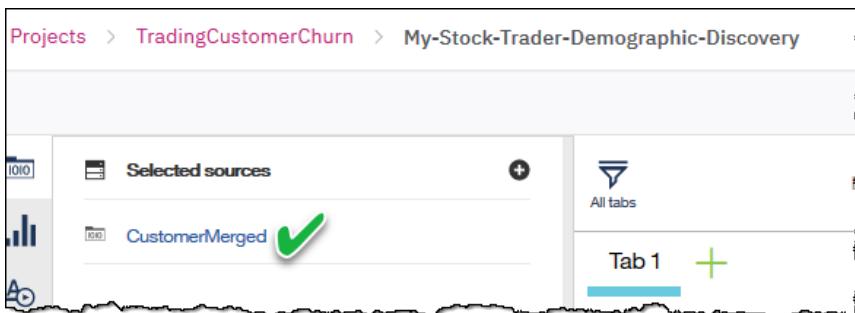
_217. In the top right portion of the screen, click **Remote** on the flyout menu.

Expand **CustomerMerged** using the down arrow and the click **Insert to dashboard**.

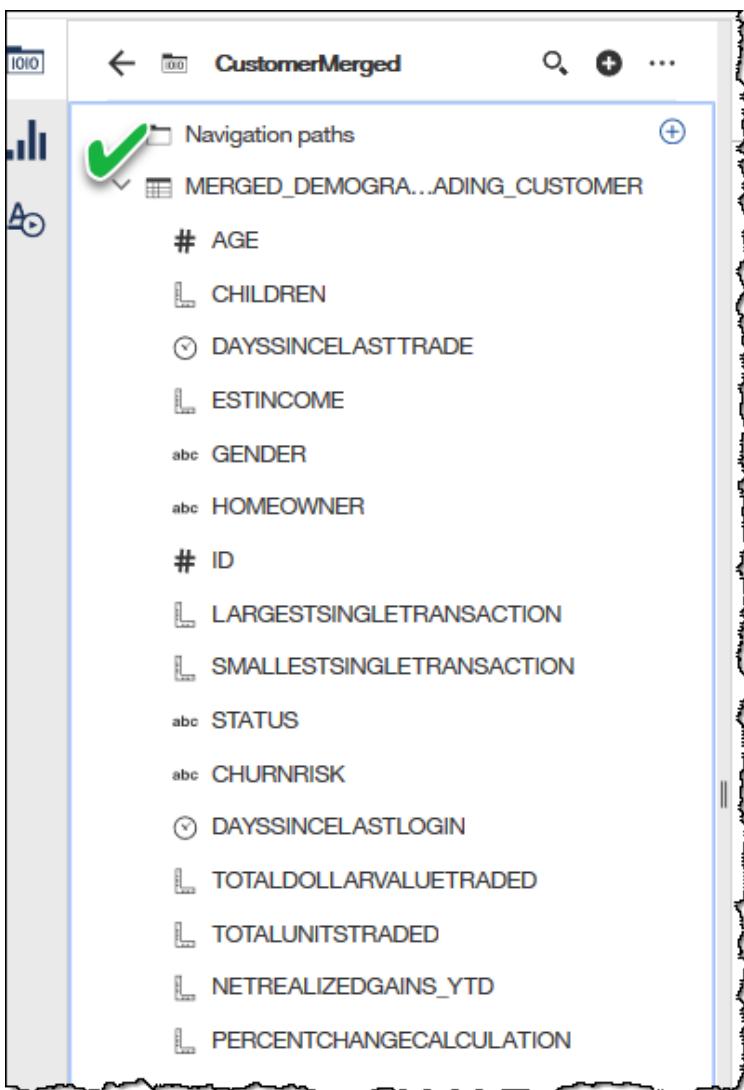


- _218. Notice in the top left of your screen a data set named [CustomerMerged](#), is inserted in the [Selected sources](#) box. This is actually a name representing a table we saw earlier in the Db2 Warehouse when we were reviewing the connection to that database.

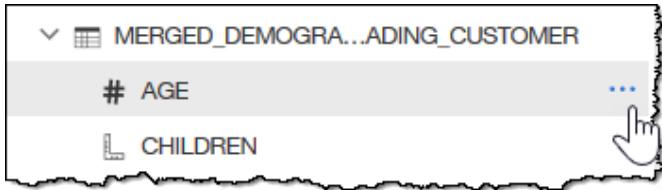
Click [CustomerMerged](#) to select it for use in our dashboard.



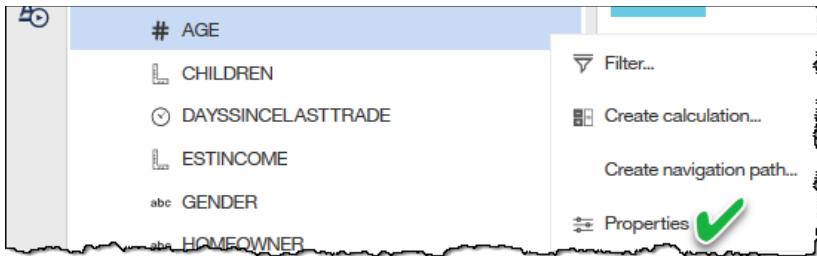
- _219. Expand the table details by clicking the twistie next to the table name [MERGED_DEMOGRAPHICS_TRADING_CUSTOMER](#).



__220. Click **Age** and then click ellipse (right side).

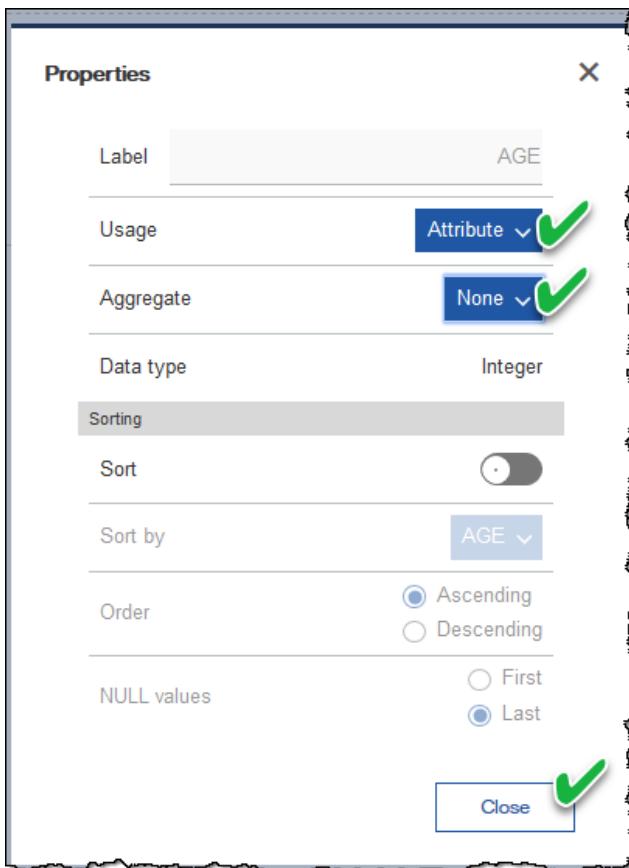


__221. Click **Properties**.



__222. Select **Attribute** from the *Usage* drop-down menu and **None** from the *Aggregate* drop-down menu.

Click **Close** to finish the selection.



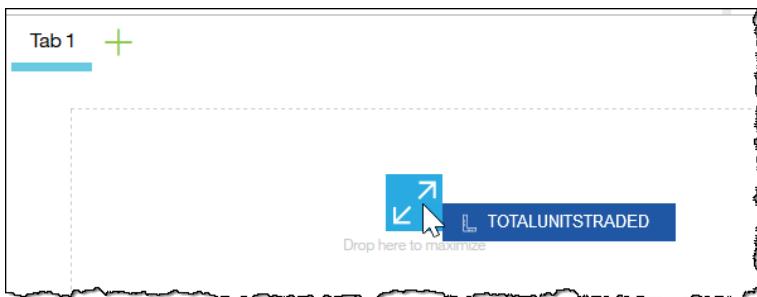
- __223. Repeat same for all columns in this table, setting their **Usage** and **Aggregate** values as shown below. Note - the key ones that can affect the dashboard are in **red**, be sure to get those correct.

Column Name	Usage	Aggregate
AGE	Attribute	None
CHILDREN	Attribute	None
DAYSSINCELASTTRADE	Measure	None
ESTINCOME	Measure	None
GENDER	Attribute	None
HOMEOWNER	Attribute	None
ID	Identifier	Count
LARGESTSINGLETRANSACTION	Measure	None
SMALLESTSINGLETRANSACTION	Measure	None
STATUS	Attribute	None
CHURNRISK	Attribute	None
DAYSSINCELASTLOGIN	Measure	None
TOTALDOLLARVALUETRADED	Measure	Total
TOTALUNITSTRADED	Measure	Total
NETRALIZEDGAINS_YTD	Measure	None
PERCENTCHANGECALCULATION	Measure	None

- __224. When you are done adjusting the properties of all the data columns, click the save icon to save the work you have done so far.



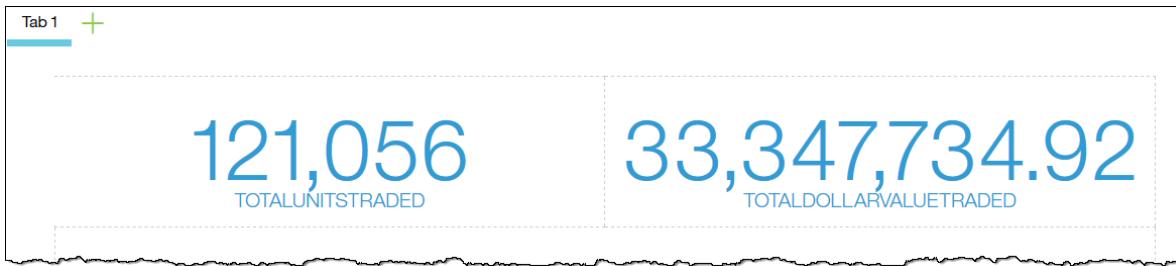
- __225. Drag **TOTALUNITSTRADED** column towards the text **Drop here to maximize** in the top left of the screen. The icon color will change to light blue when you hover over it. Drop the column there to occupy the full space of the dotted lined box in the dashboard template.



- __226. The result should look like this:



- __227. Similarly drag **TOTALDOLLARVALUETRADED** to the top right and drop it when the icon turns light blue. The result should look like this:



- __228. Now that we have scored a churn risk of existing customers in our data, let's discover the traits of those customers are by using dashboard visualizations.

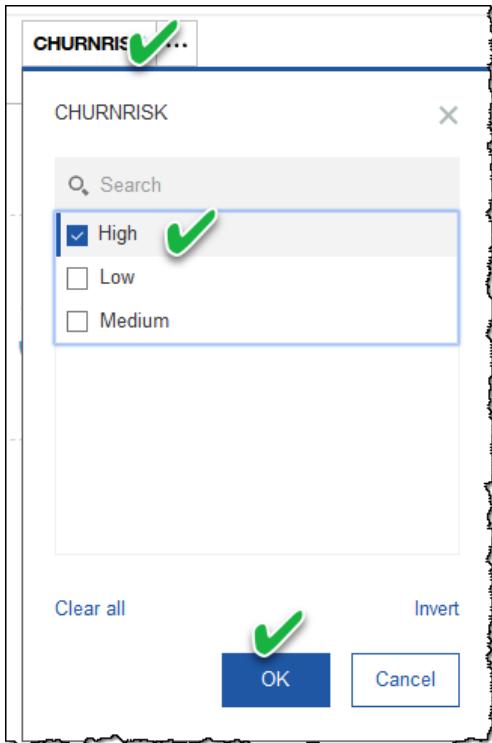
Drag **CHURNRISK** and drop it on the **This Tab** text at the top right next to the filter icon.

(Note how one is "This Tab" and the other is "All tabs". We want "This tab".)



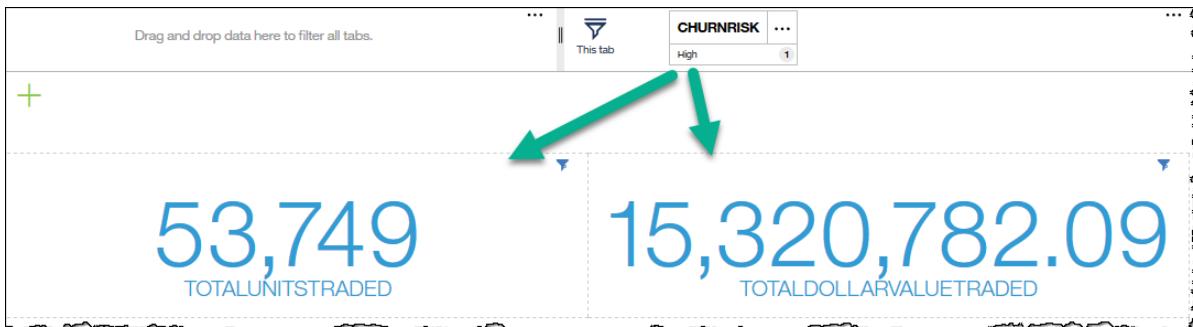
- __229. Click **CHURNRISK** to expand it and then select the box: **High**.

Click **OK** to finish this filter selection.

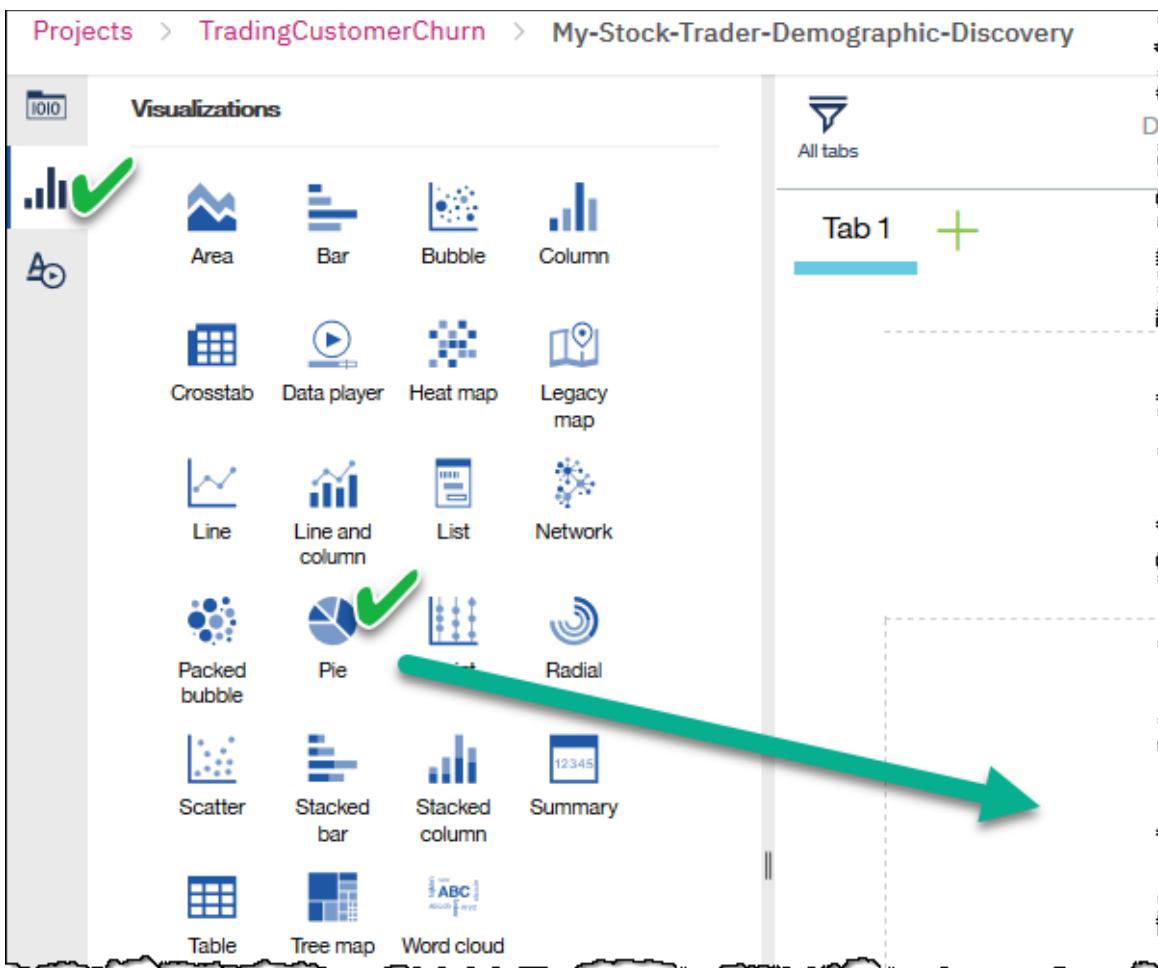


- __230. Notice the amounts change in this tab's visualizations after applying a filter of customers with a **High** churn risk. Because the filter is in box called **This tab**, all visualizations in this dashboard's tab will be affected by the filter. Other tabs created later will not be affected by this particular filter.

Each visualization in the tab now shows a tiny filter icon in the top right corner to signify that the displayed data has a filter applied to it.



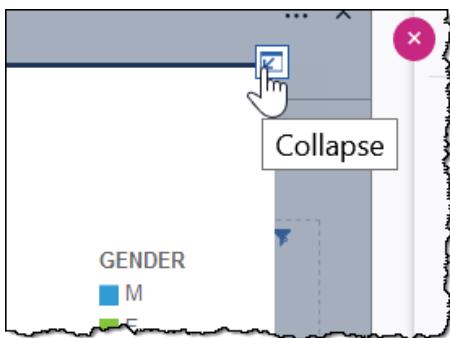
- __231. Select the Visualizations icon on the left side of your dashboard work area to expand the options for Visualizations. Choose the **Pie** chart option and drag it to bottom left side of the dashboard work area.



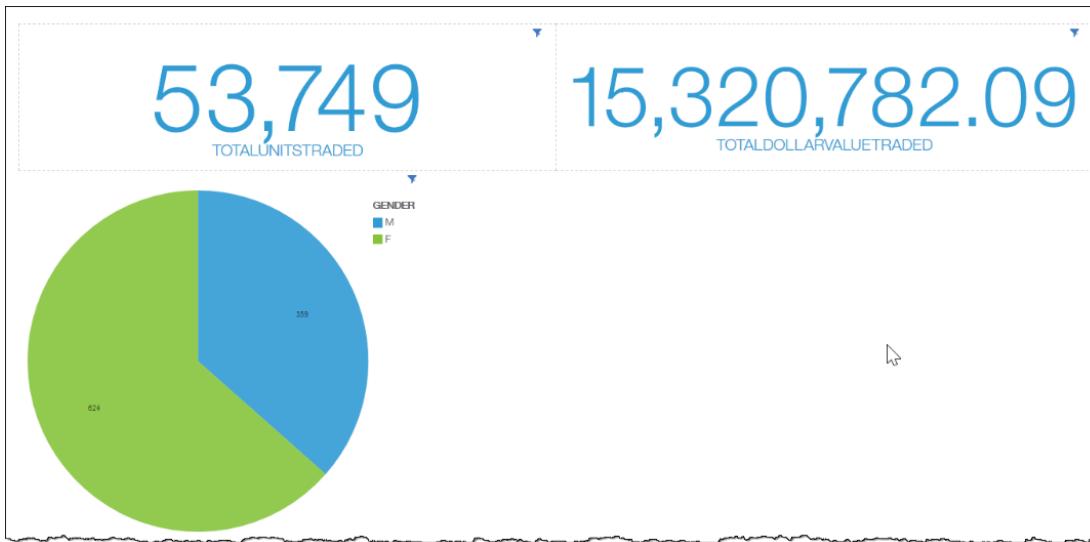
__232. The Pie chart build screen will automatically expand and allow you to begin selection of the columns required to complete the chart.

Drag GENDER to the top **Segments** section (next to the pie icon) and drag ID to the **Size #** section.

__233. Click the **Collapse** icon on the top right corner of the pie chart build screen.

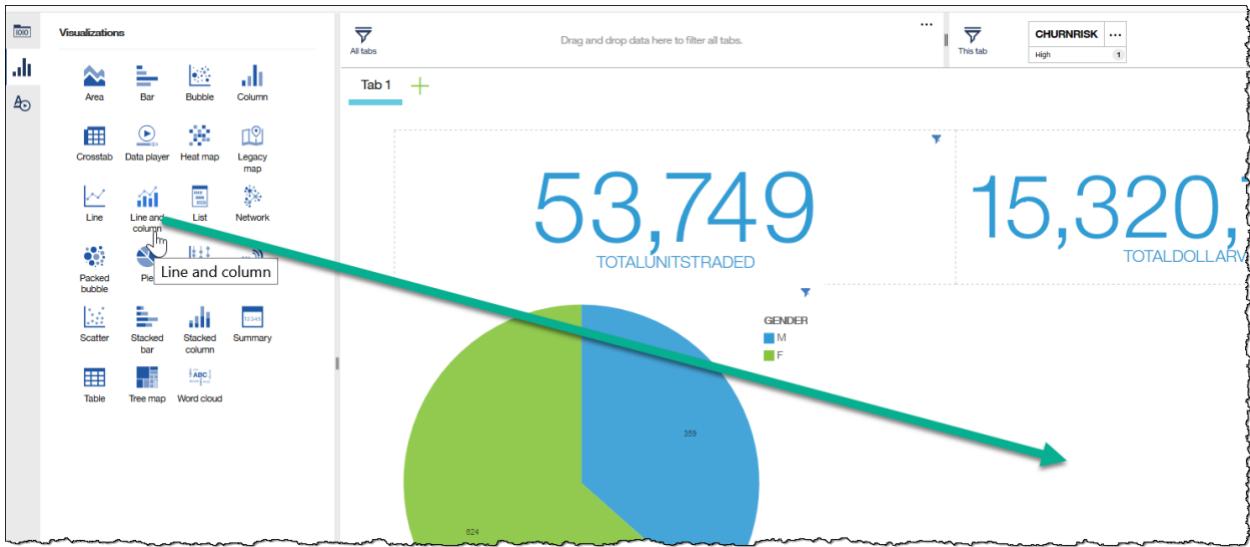


__234. Your dashboard should look something like this. If it does not, take some time to resize and move your pie chart and other visualizations to look like this.



__235. Let us now find out who makes the most trades for the most amount of trade value.

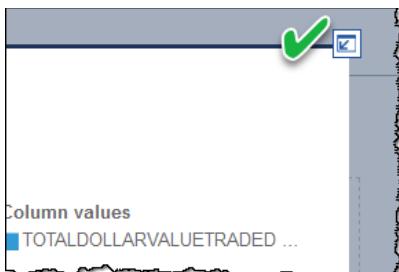
From the Visualizations options, drag **Line and Column** to the bottom right side of the work area.



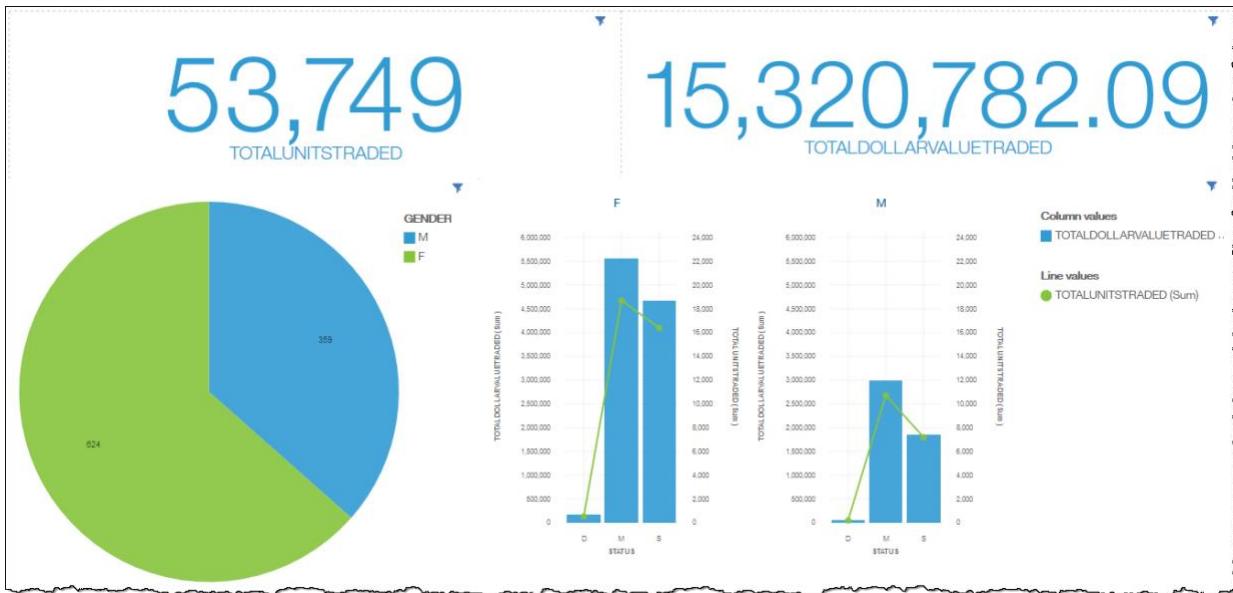
__236. Drag each of the numbered items to the right to the position indicated (drag **STATUS** to the x-axis, then **TOTALDOLLARVALUETRADED** to Length, and so on.)

The screenshot shows the IBM Watson Studio interface. On the left, the 'CustomerMerged' dataset is displayed in a tree view. Numbered circles (1, 2, 3, 4) are placed over specific items: 1 over 'STATUS', 2 over 'TOTALDOLLARVALUETRADED', 3 over 'TOTALUNITTRADED', and 4 over 'GENDER'. To the right, a configuration panel is open. It contains a list of items with corresponding slots for dragging: 'x-axis' (slot 1), '# Length *' (slot 2), '# Line position *' (slot 3), and 'Repeat (column)' (slot 4). Below this list, there is a note: '* Indicates a required field'. At the bottom, there is a general instruction: 'Drag and drop data to the slots above to build and filter the visualization.'

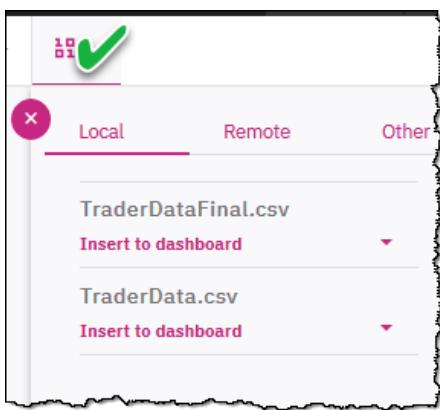
__237. Collapse the chart.



__238. The dashboard should look like below. If not, take time to resize and move the last visualization to make your dashboard like this.

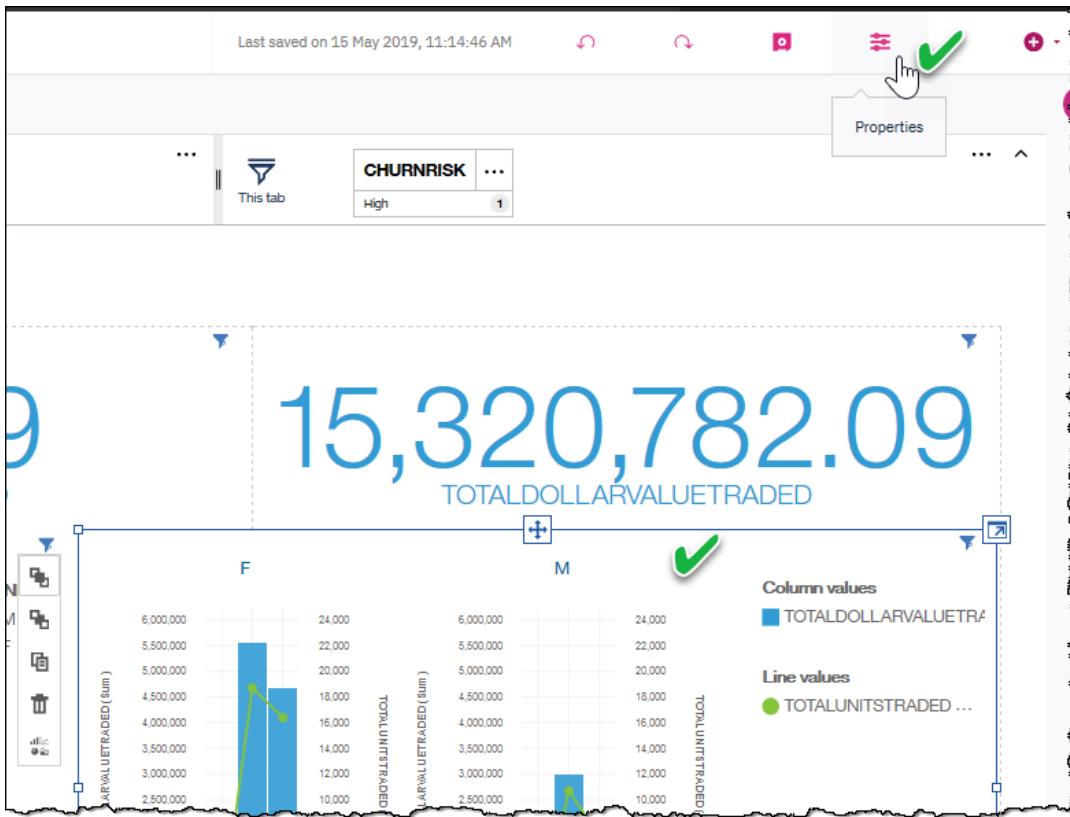


__239. You can optionally make some room in your work area by hiding the right palette by clicking the [Find Data](#) icon.

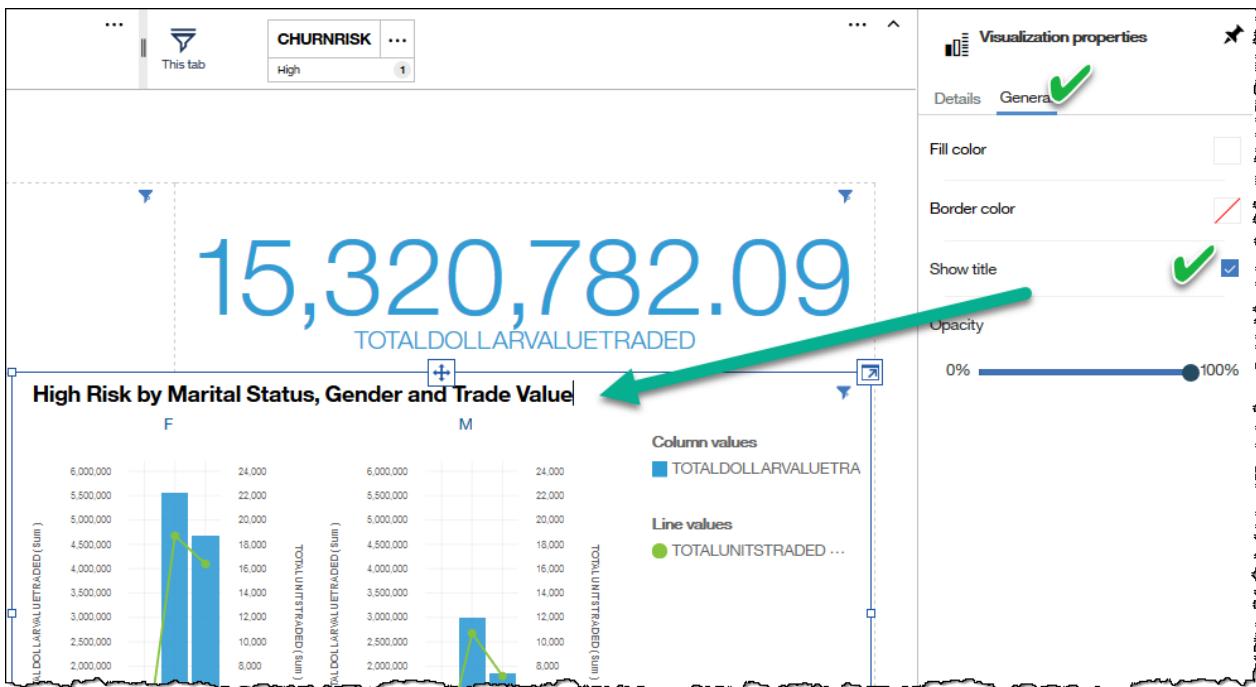


__240. Click the save icon to save the work you have done so far.

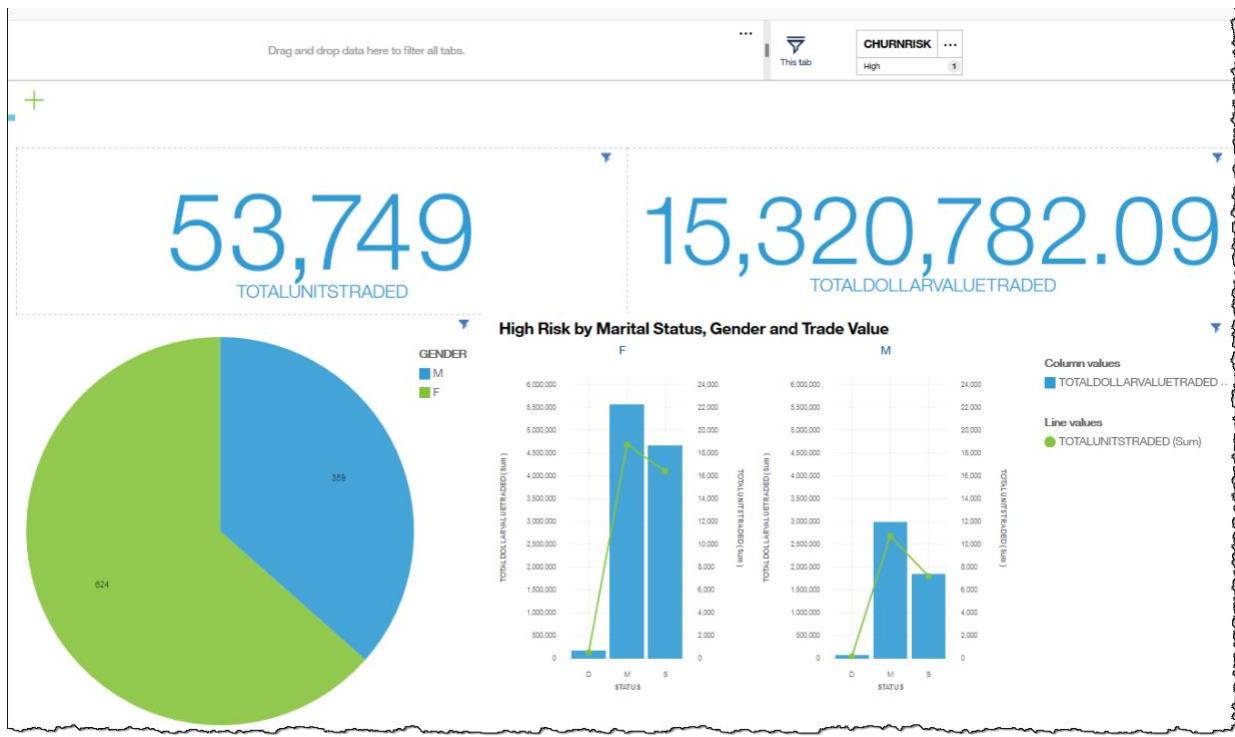
241. Click on the line and column visualization that you just created and click the **Properties** icon from top menu bar. This allows you to edit the properties of this specific visualization.



242. Click the **General** tab. Check **Show Title** and in the visualization you can now type: **High Risk by Marital Status, Gender and Trade Value**



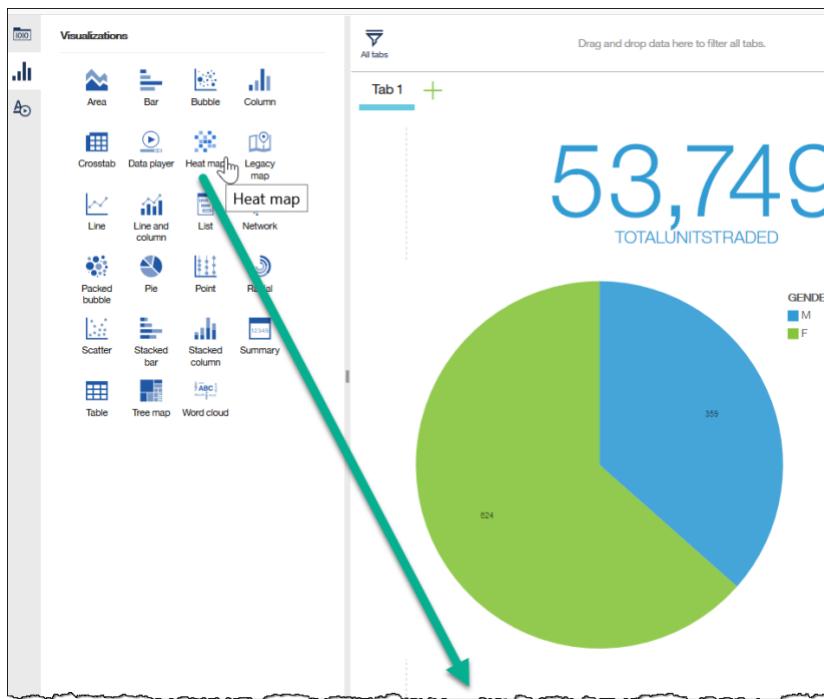
_243. Close the properties section when done. Your dashboard should look like this:



It appears from the chart that the greatest number of high-risk customers who trade and spend the most with us are Married and Single Females.

Is there any other insight we can gain? For example, does the number of children or home ownership play a part?

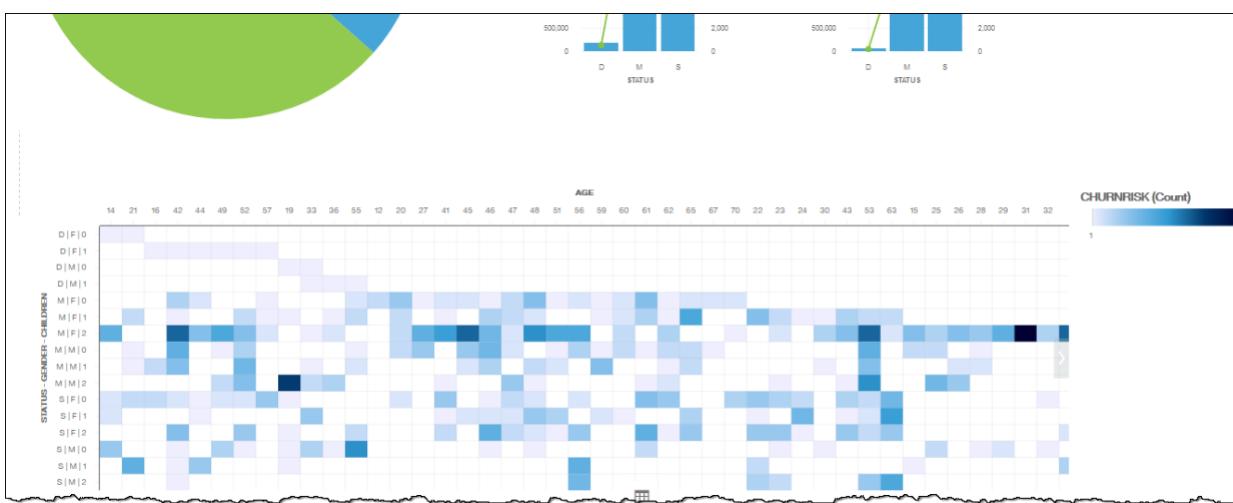
_244. From the [Visualizations](#) palette options, drag [Heat Map](#) to the bottom, below the pie chart.



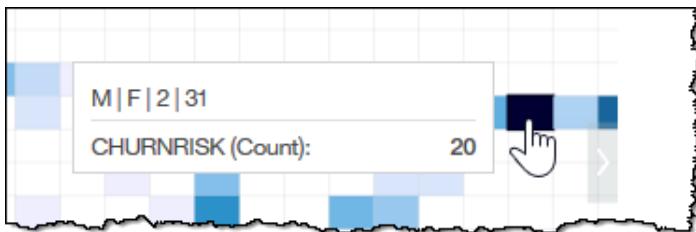
245. Drag the STATUS, GENDER, and CHILDREN individually to the Rows section, make sure to keep them in that order. Drag AGE to the Column section and drag CHURNRISK to the Heat section, as shown below

The screenshot shows the Alteryx Designer interface with the 'CustomerMerged' workflow open. On the left, the 'Navigation paths' pane lists various merged demographic fields. On the right, the 'Drag and drop data here' visualization builder is active. The 'Rows' section contains three items: STATUS (red circle 1), GENDER (red circle 2), and CHILDREN (red circle 3). The 'Columns' section contains one item: AGE (red circle 4). The 'Heat' section contains one item: CHURNRISK (red circle 5). A legend on the right maps these colors to their respective field names.

246. Collapse the Heat Map visualization and then resize and move to take up all the space below the previous visualizations.

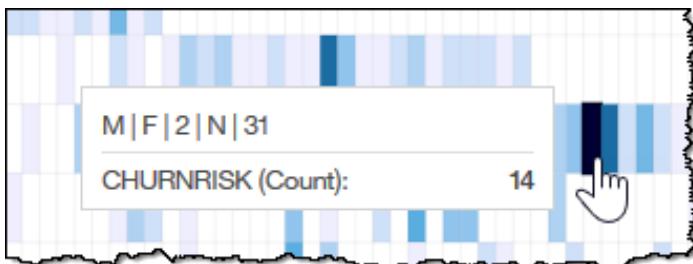


- 247. Highlighting one dark data point on this visualization (the darker, the more likely to churn) shows at risk are married females 31 years old with 2 children.



- 248. Expand the Heat Map visualization again and drag HOMEOWNER above AGE , but do not replace AGE. Collapse the visualization again. (Note: Do not replace age. Make sure to place right above it.)

- 249. Highlighting one of the darkest data points on this heatmap now shows a married female of 31 years, with 2 children and doesn't own a home has one of the greatest risks for churn at Boatswain Trading.

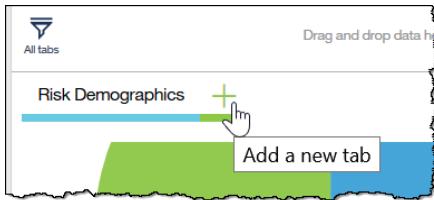


- 250. Change the title of the first tab to **Risk Demographics** By selecting the tab and then the Pencil icon to edit it.

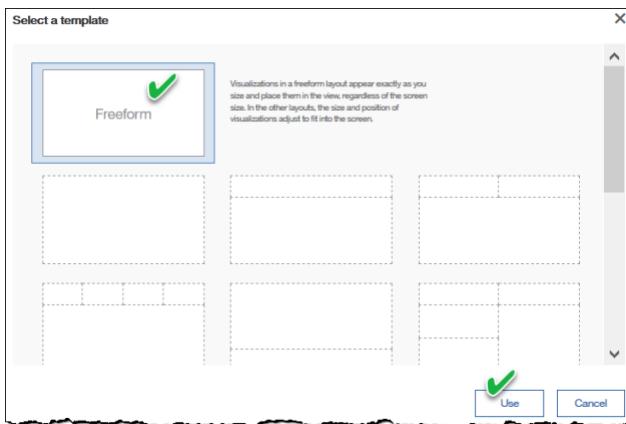
6.3.3 Analyze risk to the business

_251. Let's create another tab to see the risk to the business.

Next to the existing tab name, click **+** to add a new tab.



_252. Select the **Freeform** template and click **Use**.

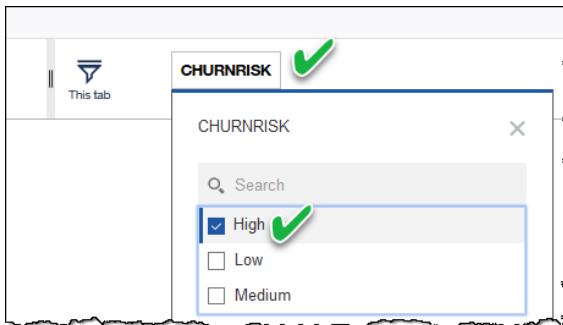


_253. Change the tab title from **Tab 1** to **Risk to the Business**.



_254. To focus on the high risk groups, we will begin by creating a series of filters

Drag column **CHURNRISK** to the top right filter area that has the text: *Drag and drop data to filter this tab*. Click it and check **High** and click **OK**.

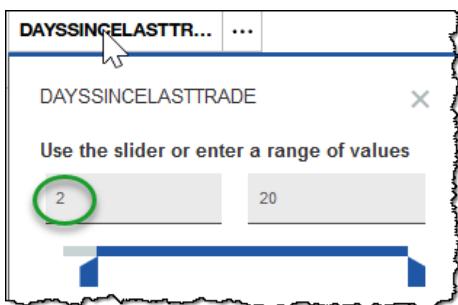


_255. Drag **GENDER** into the same filter area and check **Female**. Click **OK**.

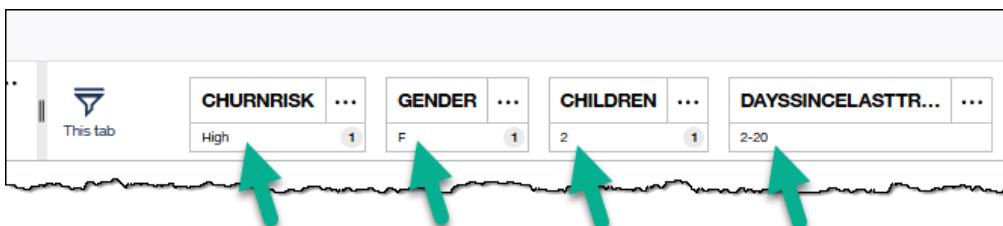
_256. Drag **CHILDREN** into the same filter area and select **2** and click **OK**.

- __257. Drag DAYSSINCELASTTRADE into the same filter area select 2 for the lower bound and keep the upper bound to 20.

Click OK.

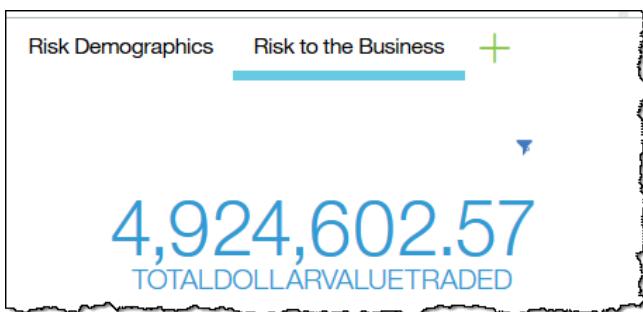


- __258. The filter area should now look like this:



- __259. Now we have applied filters that correspond to our previous findings, we can build the dashboard to drill into the details.

Drag TOTALDOLLARVALUETRADED to the top left area of the canvas. Resize the chart so it appears in the top left corner. This creates a single value field that is filtered by all the tab filters we created.

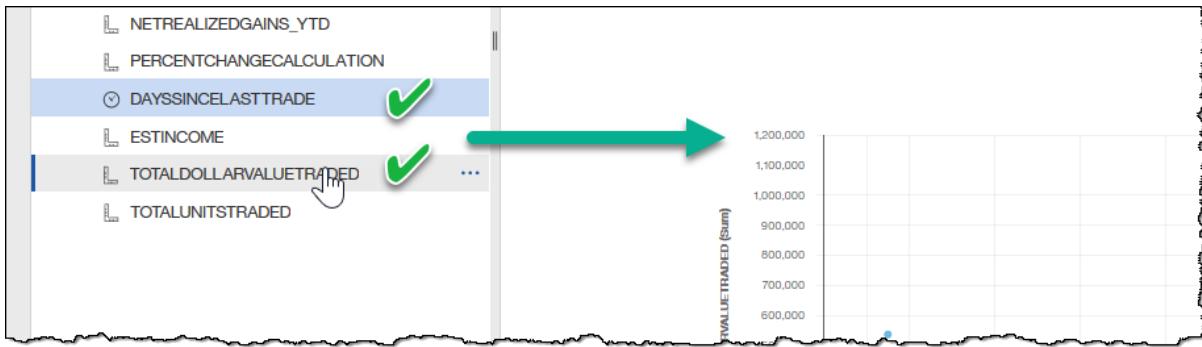


- __260. Pull over TOTALUNITSTRADED over to the top right area of the canvas. Resize the chart so that it appears in the top right corner.

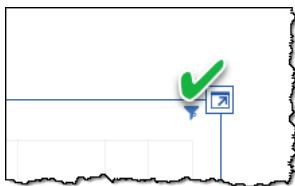


_261. Let's see the **TOTALDOLLARVALUETRADED** at risk by this group based upon the days since they last traded with us and the date they last logged in to our system.

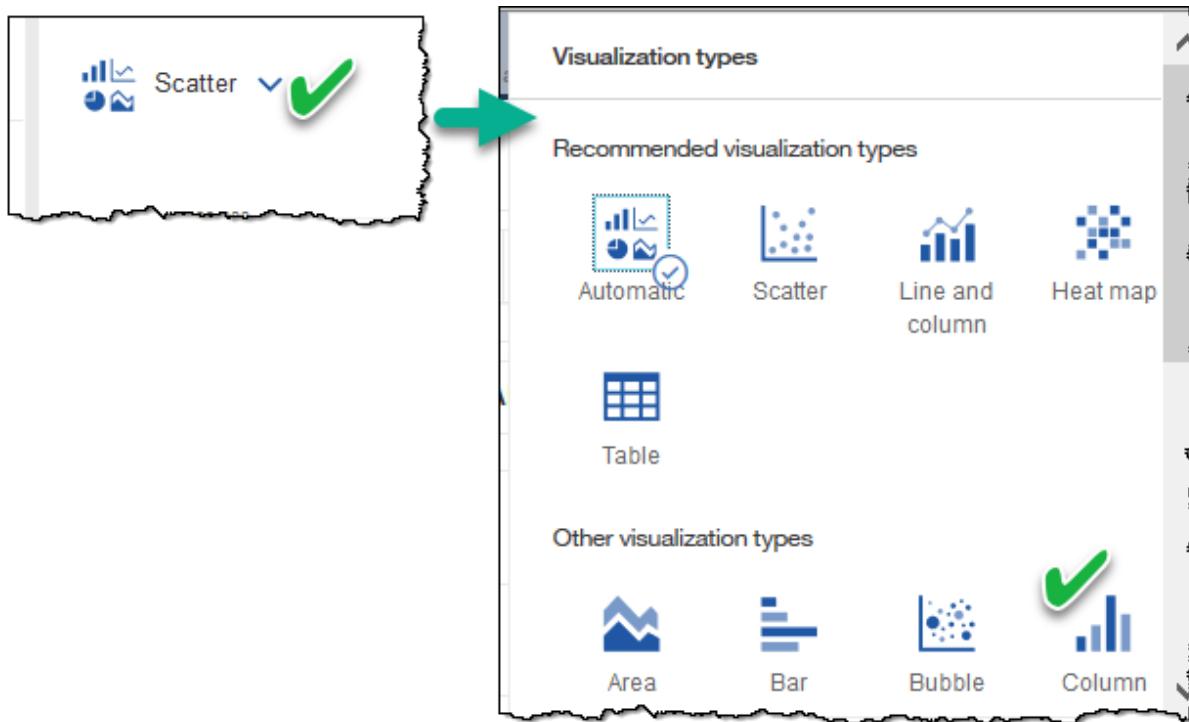
To see this, hold the control key down and click **DAYSSINCELASTTRADE** and **TOTALDOLLARVALUETRADED** and drag them both to the right.



_262. Expand the visualization that was created.



_263. Change the chart type to **Column**. [Click and select from the chart visualization.]



__264. The visualization should look like this:

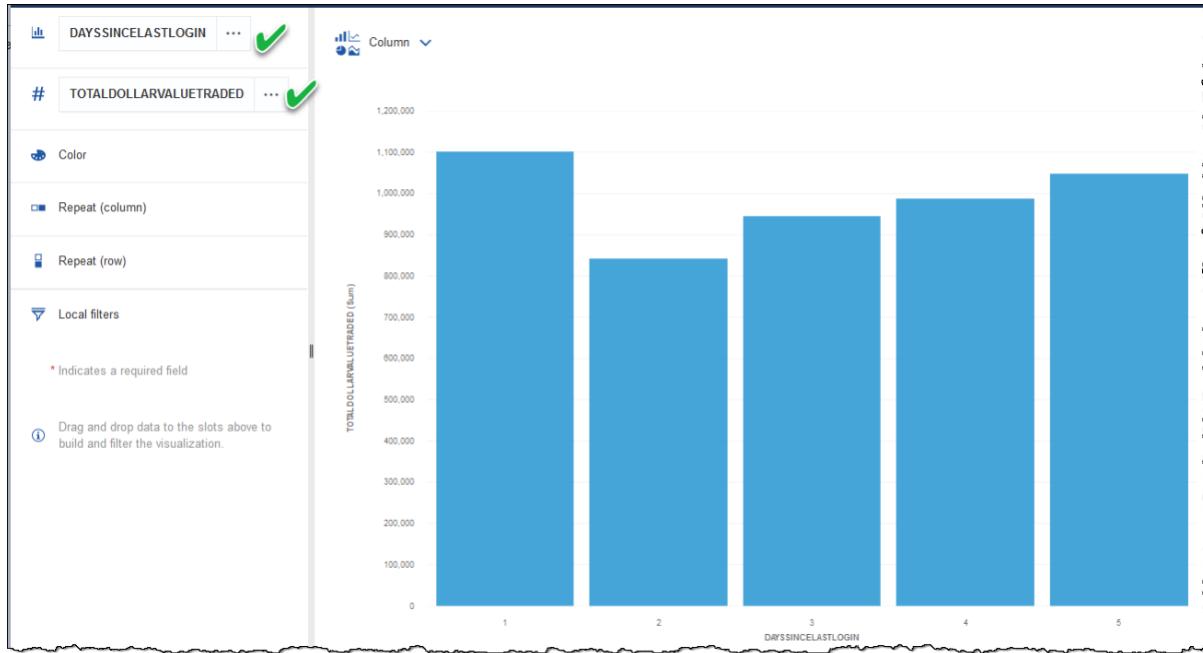


__265. Collapse the screen by clicking the collapse icon to the top right of the frame.

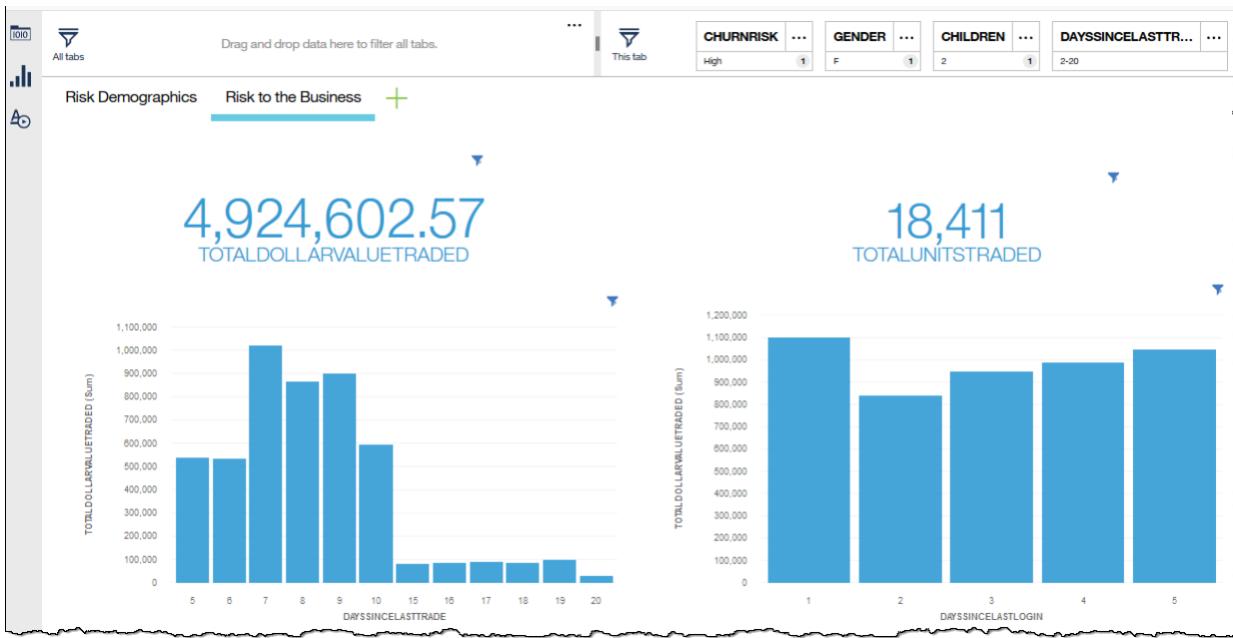
Resize this chart and fit into the bottom left of the work area. Save your work.

__266. Hold control key down and click **DAYSSINCEDLASTLOGIN** and **TOTALDOLLARVALUETRADED** and drag to the right.

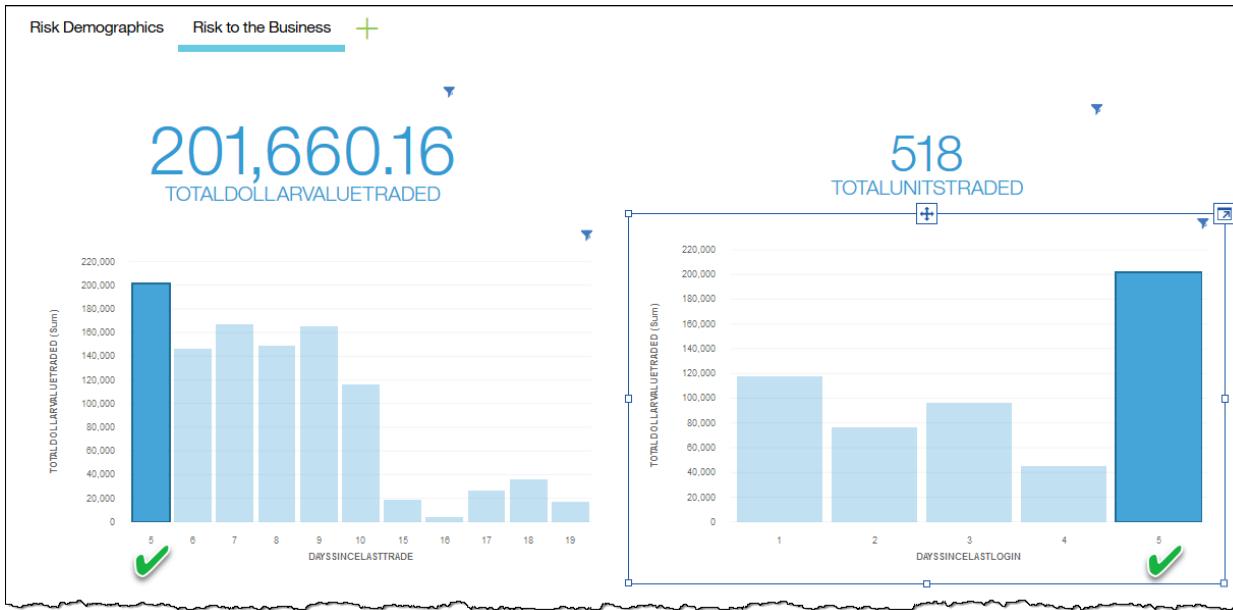
__267. Expand the chart and change the chart type to **Column**. It should display the following:



—268. Collapse this last visualization resize it and move it to make the dashboard look like this:



—269. You can click any column from each chart to show the aggregate value for total dollar traded and total units traded. For example: Click first column on first chart and last column on second chart to see the values.



Business Analyst

Playing the role of a business analyst and using these techniques, you have identified the group likely to be at risk to separate from the business.

An intelligent scientific solution needs to be crafted to mitigate the risks.

The data scientist personas are now engaged to move this analysis forward.

6.4 Lab conclusion

Dashboard creation and analysis is just one part of an Analyze phase, but a business analyst can do this relatively quickly and simply so that they can get information from your organization's data without relying on data scientists for everything.

However, the Data Scientist can take this information and create machine learning models from the knowledge gained by this kind of analysis.

In the next lab, you will explore how this is done.

**** End of Lab 06: Analyze Part 1 - Dashboards**

Lab 07 Analyze Part 2 – Model Creation

7.1 Lab overview

In the previous lab Analyze Part 1, the business analyst provided dashboards to provide insight into the data and the challenges in the organization is facing. A solution for these challenges now rests on the data scientist. These are the benefits that Cloud Pak for Data brings to this persona:

- Data search and preparation can consume up to 80% of a data scientist's time – CPD streamlines this and allows more time for the other necessary work.
- Data resides everywhere and in many formats, thus collecting all the data into one lake, or warehouse is not often realistic - CPD Data virtualization and Transformation makes this easy.
- Governing and cataloging the data gives the data scientist confidence that what they are using is secure and correct - again, CPD makes this easy.

7.2 Persona represented in this lab

The [Data Scientist](#) persona most likely one to perform the [Analyze part 2](#) tasks in this lab, that is, to create a machine learning model that can be consumed by the Stock Trader application.

Persona (Role)	Capabilities
 Data Scientist	Data Scientists brings expertise in statistics and the process of building ML/AI models to make predictions and answer key business questions.

7.3 Toolsets for analyzing data

The recent contributions of universities, research institutes and corporations to the open-source movement have led to the development of many innovative tools in data science. These include Jupyter and Zeppelin notebooks and the R and Python languages.

The default environment for Cloud Pak for Data that you will be using in this lab is:

- Jupyter Notebook Server 5.7.0
- Anaconda3 5.2 with the conda-forge channel
- Python 3.6
- Apache Spark 2.3.2

You can additionally install one or more of following development environments as add-ons to CPD:

- Jupyter Notebook Server with Python 2.7 or Python 3.5
- RStudio Server with R3.4.3
- Zeppelin Notebook Server 0.7.3 with Anaconda2 4.4

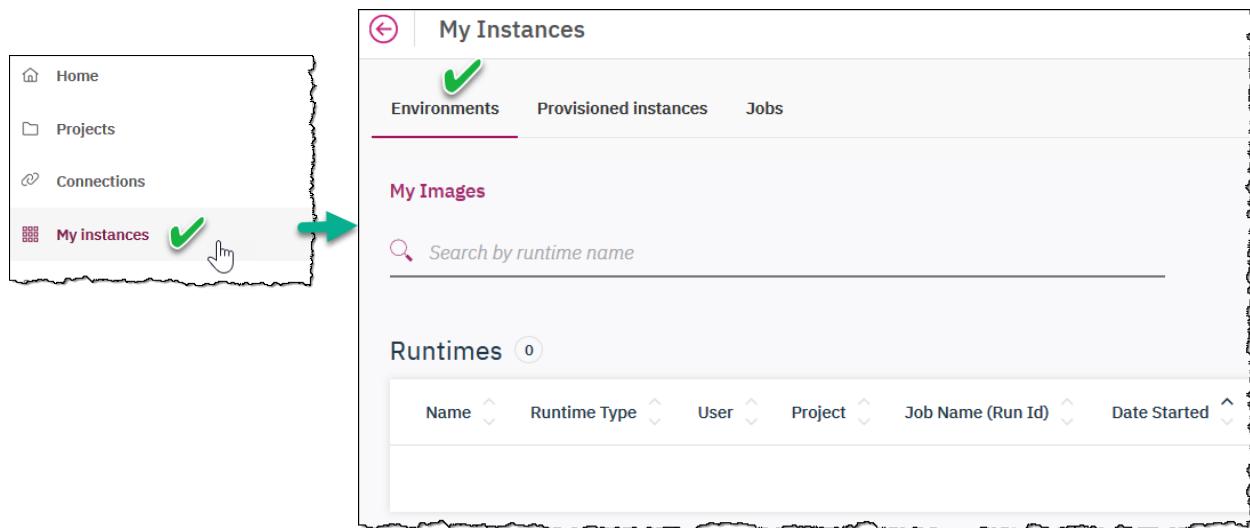
7.4 Walk through a model creation beginning steps

Developing, training, and deploying a machine learning lab from scratch is beyond the scope of this introduction to CPD, so you will walk through an existing Jupyter notebook that represents an example of the model development process. Those familiar with the process will recognize the tools and processes used in these exercises, but those who are not will get an introductory experience from it.

In this lab, you will develop, test, and score a model that addresses the business problem at hand.

- _270. Click [Navigation Menu](#) ⇒ [My Instances](#) ⇒ [Environments](#)

This location is where your provisioned Runtime environment will be able to be controlled. In the official workshop image, none should currently exist (this screen is blank) but if you were working in another environment you should check here first to review the Runtimes and potentially stop those that are running that are not being used. This is because each executing Runtime environment takes up compute resources on your CPD cluster.



- _271. From the [Navigation Menu](#) ⇒ [Projects](#)

- _272. Click [TradingCustomerChurn](#) to open it.

Projects			
<i>Search</i>		All projects	
Name	Project Type	User Role	Last Updated
TradingCustomerChurn	Analytics	Admin	22 May 2019, 3:57 PM

_273. Under **Assets** tab, click **Notebooks**

The screenshot shows the 'Assets' tab selected in the top navigation bar. Below it, the 'Notebooks' section is highlighted with a green checkmark. A search bar is present above the notebook list. The notebook '01TradingCustomerChurnClassifierSparkML' is listed with a green checkmark next to its name.

_274. Click the ellipsis next to Notebook **01TradingCustomerChurnClassifierSparkML**

Choose **Open in Jupyter with Python 3.6, Spark 2.3.2**

The screenshot shows the 'Notebooks' list with one item: '01TradingCustomerChurnClassifierSparkML'. A context menu is open for this item, listing 'Preview', 'Open in Jupyter with Python 3.6, Spark 2.3.2' (which has a green checkmark and a hand cursor icon), and 'Publish'.

 Data Scientist	<p>The Jupyter notebook process initiates a request to Kubernetes to launch a pod that will run on any node in the CPD cluster where sufficient resources are available.</p> <p>The launch may take a minute or so because it takes time to load the 7 GB image from the Docker registry and then launch it.</p>
---	--

- ___275. If you see a message like this one, **X** out of it for now.



- ___276. A notebook is a collection of compute cells that can be run individually, or as a group. You can pause execution between steps to review its results, make changes to your code, and re-run the cell. One of the features of notebooks is that markdown cells (notes) can be interspersed with code which allows notebooks to be self-documenting.

A notebook can use the environment of your choice. For example, we chose to use Python 3.6 with Spark 3.2.3 because it is the default for the current version of CPD. Others are available per the “toolsets” overview at the beginning of this lab.

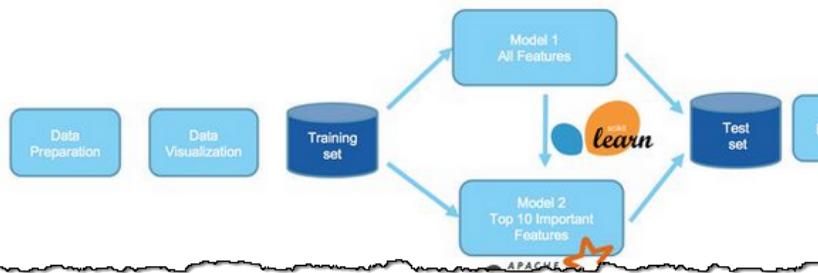
A screenshot of a Jupyter Notebook interface. The title bar shows "TradingCustomerChurn > Notebooks > 01TradingCustomerChurnClassifierSparkML". The toolbar includes File, Edit, View, Insert, Cell, Kernel, Widgets, Help, and various icons for file operations. A green callout bubble points to the "Run" button in the toolbar with the text "Runs the cell you are positioned on". Another green callout bubble points to the left edge of the first cell with the text "Denotes the cell you are positioned on". The notebook content includes a section titled "Trading Platform" and "Customer Churn Prediction using SparkML". The text discusses the goal of running analytics and predicting churn based on user activity while maintaining profitability. It also mentions leveraging Data Science Experience Local to build a classification model using the SparkML library.

277. Read through the introduction to learn what this notebook provides.

The notebook explains two main types of cells: [markdown](#) and [code](#). Markdown cells are for documentation (the first cells in this notebook) and code cells that execute the Notebook processes to produce results.

In this notebook, we will leverage Data Science Experience Local to do the following:

1. Ingest merged customer demographics and trading activity data
2. Visualize merged dataset and get better understanding of data to build hypotheses for prediction
3. Leverage SparkML library to build classification model that predicts whether customer has propensity to churn
4. Expose SparkML classification model as RESTful API endpoint for the end-to-end customer churn risk predict

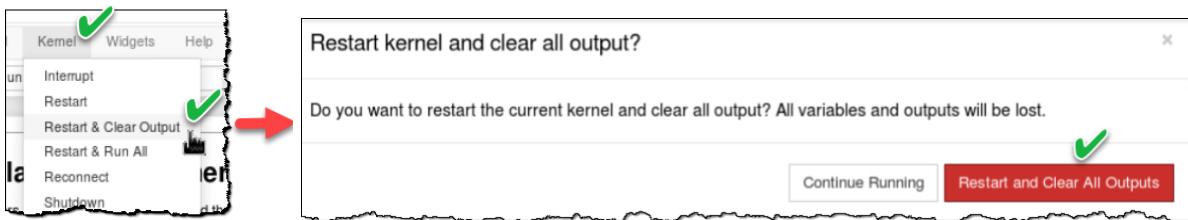


Quick set of instructions to work through the notebook

If you are new to Notebooks, here's a quick overview of how to work in this environment:

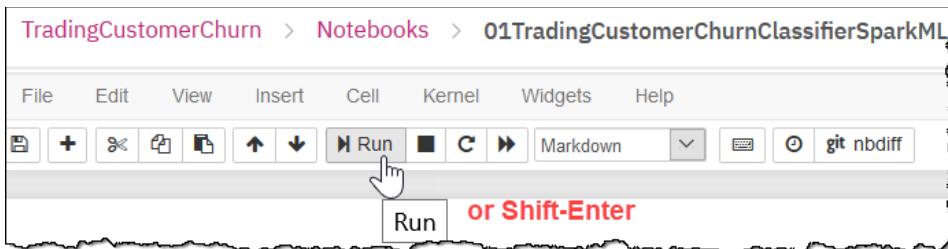
1. The notebook has 2 types of cells - markdown (text) such as this and code such as
2. Each cell with code can be executed independently or together (see options under
3. To run the cell, position cursor in the code cell and click the Run (arrow) icon. The
4. Work through this notebook by reading the instructions and executing code cell by

_278. Click **Kernel** \Rightarrow **Restart & Clear Output**. When prompted, click on the action again.



Wait for about 15 seconds after you do this for the kernel restart to take effect.

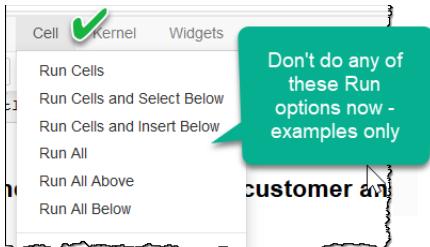
_279. Note: To execute a cells (including markdown cells) you can either click the menu icon **Run** or use keyboard **Shift-Enter** execute the cell you are positioned on.



Click **Run** (or **Shift-Enter**) to execute the cell first cell you are positioned on. Note that executing a markdown cell essentially skips it because nothing happens to that cell.

You can execute cell one at a time, in which case execution pauses and positions at the next cell.

You could also "Run All" cells which causes a serial execution from cell to cell without pause. (Do not do this because you will want to review each cell as you go through this Notebook.)



280. Execute the first few cells in this Notebook until you reach this first code cell. Run it to load the libraries you will need for the rest of code cells in this Notebook.

1. Load libraries [¶](#)

[Top](#)

Running the following cell will load all libraries needed to load, visualize, prepare the data and build ML models for our use case

```
In [1]: import os
from pyspark.sql import SQLContext
from pyspark.sql.types import DoubleType
from pyspark.ml.feature import OneHotEncoder, StringIndexer, VectorIndexer, IndexToString
from pyspark.sql.types import IntegerType
from pyspark.ml import Pipeline
from pyspark.ml.feature import VectorAssembler
from pyspark.ml.classification import RandomForestClassifier, NaiveBayes
from pyspark.ml.evaluation import MulticlassClassificationEvaluator
from pyspark.mllib.evaluation import MulticlassMetrics
import brunel
from dss_ml.ml import save
import pandas as pd
import numpy as np
```

Notice that when a code cell is executed, the Notebook automatically numbers them each time this occurs. This helps tell what order the cells have been executed in.

In [1]: [¶](#)

Denotes code cell numbering

```
import os
from pyspark.sql import SQLContext
from pyspark.sql.types import DoubleType
from pyspark.ml.feature import OneHotEncoder, StringIndexer, VectorIndexer, IndexToString
from pyspark.sql.types import IntegerType
from pyspark.ml import Pipeline
from pyspark.ml.feature import VectorAssembler
from pyspark.ml.classification import RandomForestClassifier
from pyspark.ml.evaluation import MulticlassClassificationEvaluator
from pyspark.mllib.evaluation import MulticlassMetrics
import brunel
from dss_ml.ml import save
import pandas as pd
import numpy as np
```

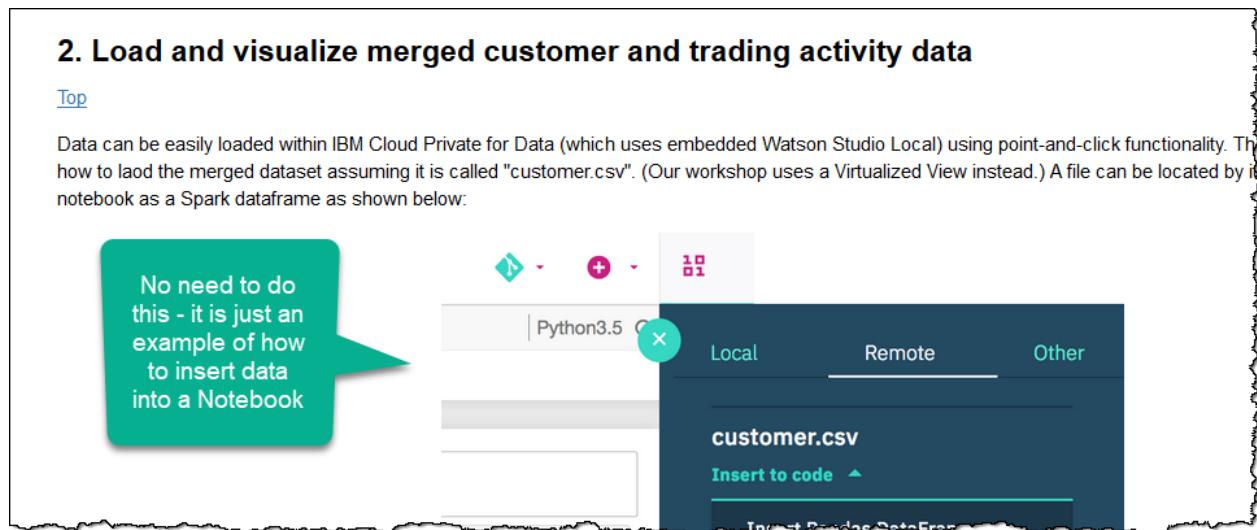
- 281. The next markdown cell gives an example of the procedure to insert a Panda or Spark data frame. Ignore these instructions because we have already inserted the code to connect to the data.

[Run](#) this markdown cell to continue to the next code cell.

2. Load and visualize merged customer and trading activity data

[Top](#)

Data can be easily loaded within IBM Cloud Private for Data (which uses embedded Watson Studio Local) using point-and-click functionality. This how to load the merged dataset assuming it is called "customer.csv". (Our workshop uses a Virtualized View instead.) A file can be located by its notebook as a Spark dataframe as shown below:



- 282. The next cell is the second code cell and shows how a Notebook connects to a data source. It should look like the example below. This code has the key coding part circled that need to be correct for the rest of the notebook to run.

If the virtualized view you created is not the same name as the one shown here, change it in the cell before running it.

```
# Add asset from remote connection
df_churn = None
dataset = dsx_core_utils.get_remote_data_set_info('USER999.DEMOCHURN_ACTIVITY_VIRTUALIZED')
dataSource = dsx_core_utils.get_data_source_info(dataset['datasource'])
sparkSession = SparkSession(sc).builder.getOrCreate()
# Load JDBC data to Spark dataframe
dbTableOrQuery = (dataSet['schema'] + '.' if(len(dataSet['schema'].strip()) != 0) else '') + dataSet['name']
df_churn = sparkSession.read.format("jdbc").option("url", dataSource['URL']).option("dbtable", dbTableOrQuery).load()
df_churn.show(5)
```

- __283. Click [Run](#) (or [Shift-Enter](#)). While the cell is running, you will see the **[*]** indicator (meaning it is running.)

```
In [*]: M # Add asset from remote co
          import dax.core.utils, requests, os, io
          from pyspark import SparkSession
          from dax.core import connection
          database = dax.core.utils.get_remote_data
          source = "dax.core.utils.get_remote_data"
          source
```

- __284. The output will look like this which should be recognizable as a select of the first 5 rows from the virtualized view you created earlier: [USER999.DEMOCHURN_ACTIVITY_VIRTUALIZED](#).

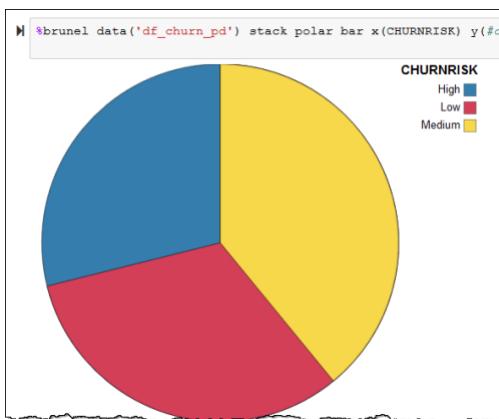
AGE	AGE_GROUP	CHILDREN	CHURNRISK	ESTIMATED_INCOME	GENDER	HOME_OWNER	ID	STATUS	DAYSSINCELASTLOGIN	DAYSSINCELASTTRADE	ENTCHANGECALCULATION	SMALLEST_SINGLETRANSACTION	TOTALDOLLARVALUETRADED	TOTALUNITSTRADED	_ID
11.25	Adult	2	Low	29616.00	M	N	1	M	45	5CE5A68902F71509E...	3	1489.149	29782.98		
5.5	Adult	0	Low	19732.80	M	N	2	M	22	5CE5A68902F71509E...	1	1240.624	24812.48		
8.0	Adult	2	High	96.33	M	N	3	S	32	5CE5A68902F71509E...	3	1306.6305	26132.61		
3.45	Adult	2	High	52004.80	F	N	4	M	23	5CE5A68902F71509E...	2	125.7625	5030.5		
11.5	Adult	2	High	53010.80	M	N	5	M	46	5CE5A68902F71509E...	2	622.5625	12451.25		

only showing top 5 rows

 Data Scientist	<p>You do not have to write the code to use data in a Notebook. The Insert to code capability shown in the previous markdown cell can do this for you.</p>
--	--

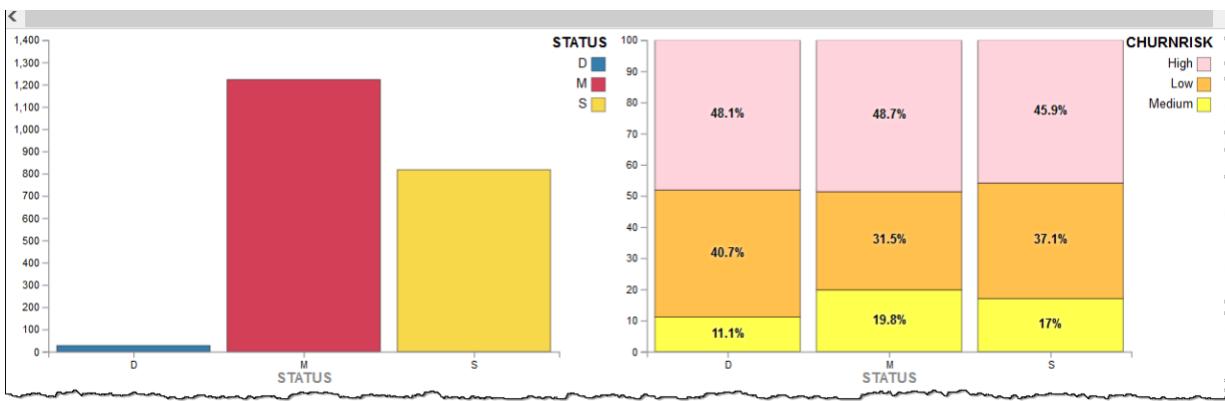
- __285. The next cell contains code to visualize the data using the Brunel library.

[Run](#) the cell.

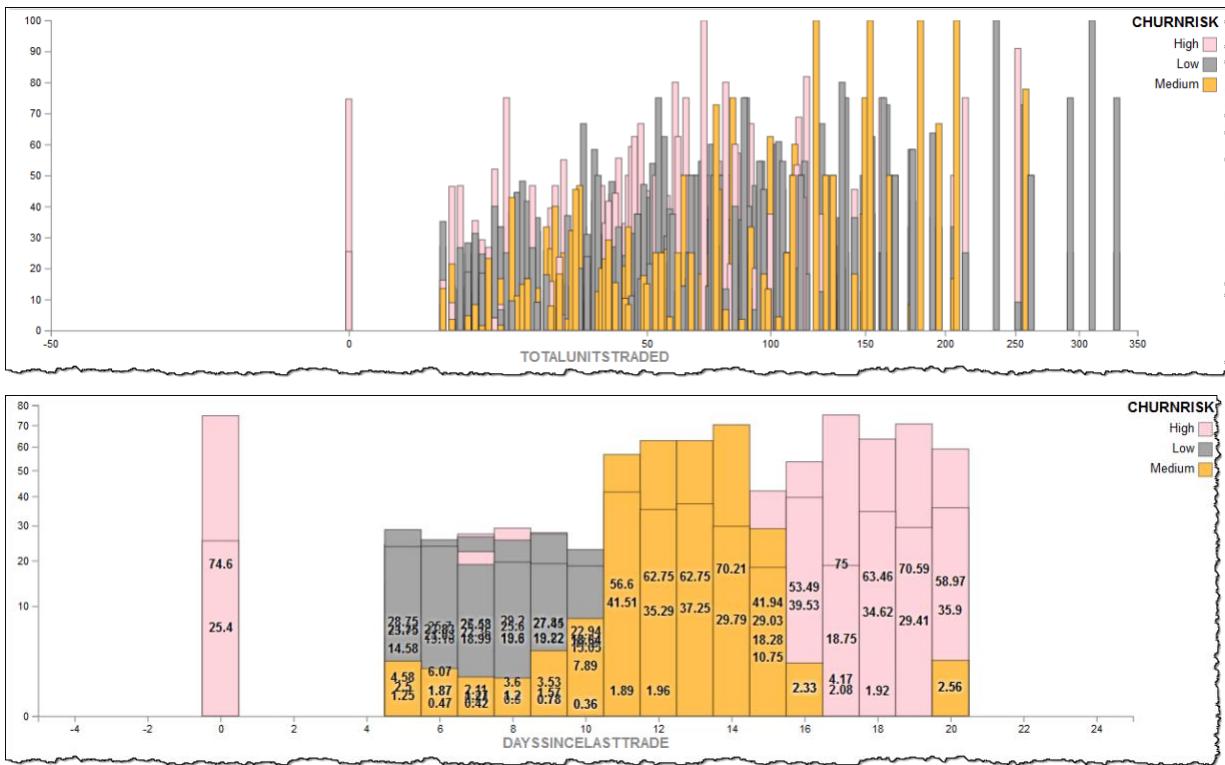


Brunel is just one of many open source visualization libraries available for Python and Jupyter. There are others like matplotlib, plotly, and Pixiedust.

__286. Run the next cell to show a visualization of CHURNRISK by different categories and status.



__287. Run the two next cells to see more visualizations breaking down CHURNRISK.



_288. Run the next cells to prepare the data for the model development.

Read through the first markdown cell in this series to get an understanding of what these next steps involve.

3. Data preparation

[Top](#)

Data preparation is a very important step in machine learning model building. This is because the mod

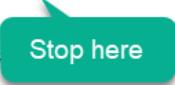
During this process, we identify categorical columns in the dataset. Categories needed to be indexed, indicating the presence of a specific feature value from among the set of all feature values. This encod

Final step in the data preparation process is to assemble all the categorical and non-categorical column

_289. Stop at the markdown cell that says [Build SparkML Random Forest classification model](#)

4. Build SparkML Random Forest classification model

[Top](#)

 Stop here

7.4.1 Build a Spark ML model using Random Forest Classification

_290. Read through the markdown cell to understand what you are doing next.

4. Build SparkML Random Forest classification model

[Top](#)

We instantiate a decision-tree based classification algorithm, namely, `RandomForestClassifier`. Next we define a pipeline to cha learning algorithms to make it easier to combine multiple algorithms into a single pipeline, or workflow.

We split original dataset into train and test datasets. We fit the pipeline to training data and apply the trained model to transform

- __291. Run the next two code cells that define the classifier and index labels and define the train and test datasets for a pipeline chain.

```

|M # instantiate a random forest classifier, take the default settings
rf=RandomForestClassifier(labelCol="label", featuresCol="features")

# Convert indexed labels back to original labels.
labelConverter = IndexToString(inputCol="prediction", outputCol="predictedLabel", labels=labelIndexer.labels)

stages += [labelIndexer, assembler, rf, labelConverter]

pipeline = Pipeline(stages = stages)

|M # Split data into train and test datasets
train, test = df_churn.randomSplit([0.7, 0.3], seed=100)
train.cache()
test.cache()

]: DataFrame[AGE: int, AGE_GROUP: int, CHILDREN: int, CHURNRISK: string, ESTIMATED_INCOME: int, GENDER: string,
STTRADE: int, LARGESTSINGLETRANSACTION: int, NETREALIZEDGAINS_YTD: int, NETREALIZEDLOSSES_YTD: int, PERCENT,
TOTALUNITSTRADED: int, ID: int]

```

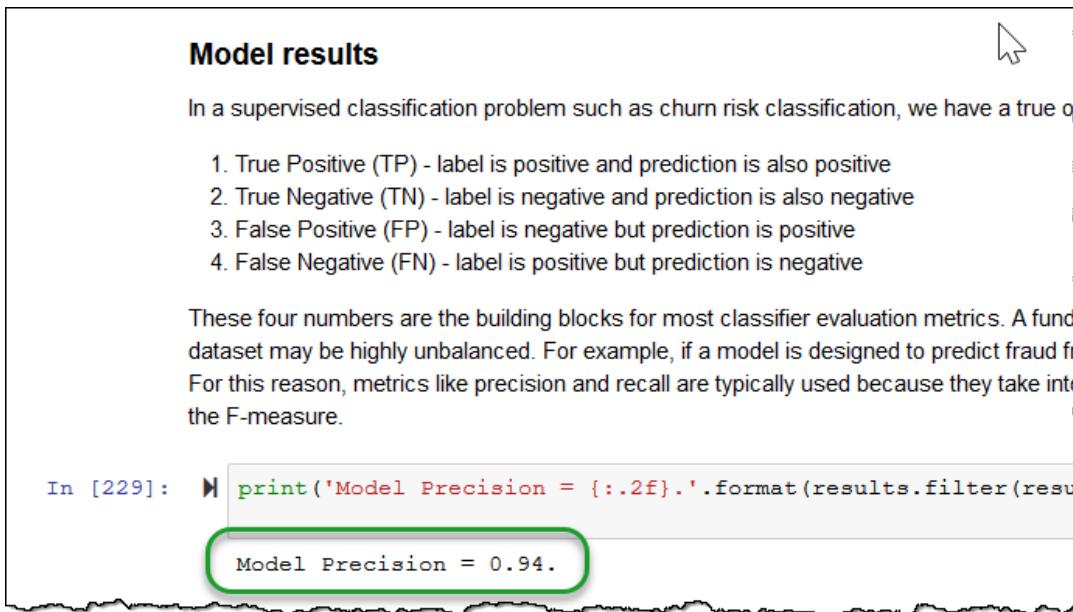
- __292. Run the next few cells to build the model using the pipeline, then transform and show the results of the likelihood of churn for a particular customer (by ID).

	ID	CHURNRISK	label	predictedLabel	prediction	probability
0	1044	Medium	2.0	Medium	2.0	[0.18256265483446013, 0.0, 0.8174373451655399]
1	1631	High	0.0	High	0.0	[0.961111111111111, 0.0, 0.03888888888888888]
2	1754	High	0.0	High	0.0	[0.961111111111111, 0.0, 0.03888888888888888]
3	1715	High	0.0	High	0.0	[0.8960250741768074, 0.021179039301310047, 0.0...]
4	1310	Low	1.0	Low	1.0	[0.0, 0.9921115821527794, 0.007888417847220615]
5	1072	Low	1.0	Low	1.0	[0.02620087336244542, 0.9636144206985578, 0.01...

- __293. Run the cells through the Model results markdown cell

__294. The “Model results” markdown cell describes in detail how a model like this is classified.

Run the next cell to see the Model Precision. (Note: your results may vary.)



Model results

In a supervised classification problem such as churn risk classification, we have a true o

1. True Positive (TP) - label is positive and prediction is also positive
2. True Negative (TN) - label is negative and prediction is also negative
3. False Positive (FP) - label is negative but prediction is positive
4. False Negative (FN) - label is positive but prediction is negative

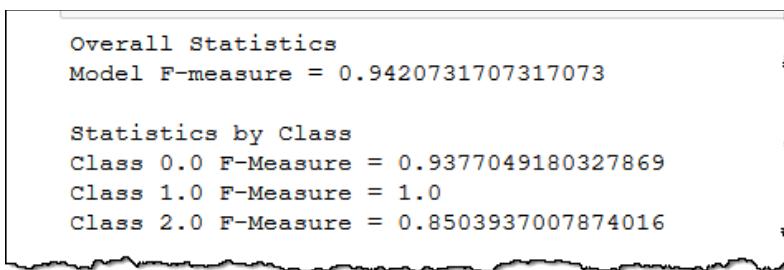
These four numbers are the building blocks for most classifier evaluation metrics. A fund dataset may be highly unbalanced. For example, if a model is designed to predict fraud fr For this reason, metrics like precision and recall are typically used because they take into the F-measure.

```
In [229]: print("Model Precision = {:.2f}.".format(results.filter(resu
Model Precision = 0.94.
```

7.4.2 Evaluate different models

__295. The process of building model is incomplete unless it can be evaluated with a test data to see its accuracy.

Run until you see output similar to this:



```
Overall Statistics
Model F-measure = 0.9420731707317073

Statistics by Class
Class 0.0 F-Measure = 0.9377049180327869
Class 1.0 F-Measure = 1.0
Class 2.0 F-Measure = 0.8503937007874016
```

__296. Next is Naïve Bayes classifier trained on the training data set. Notice this is not as good as the Random Forest classifier that shows a high F-measure.

Run until you see output similar to this:

```
Before we save the random forest classifier to repository, let us first evaluate the performance of a simple Naive Bayes classifier trained on the training dataset.

[50]: nb=NaiveBayes(labelCol="label", featuresCol="features")

stages_nb = stages
stages_nb[-2] = nb

pipeline_nb = Pipeline(stages = stages_nb)

# Build models
model_nb = pipeline_nb.fit(train)
results_nb = model_nb.transform(test)

print('Naive Bayes Model Precision = {:.2f}'.format(results_nb.filter(results_nb.label == results_nb.prediction)))
Naive Bayes Model Precision = 0.71.
```

7.4.3 Save the model in the repository

__297. After a comparison of two models, the Random Forest model is saved into the repository.

```
save(name='TradingChurnRiskClassificationSparkML',
      model=model,
      test_data = test,
      algorithm_type='Classification',
      description='This is a SparkML Model to Classify Trading Customer Churn Risk')

{'path': '/user-home/999/DSX_Projects/TradingCustomerChurn/models/TradingChurnRiskClassificationSparkML/1',
 'scoring_endpoint': 'https://dsxl-api/v3/project/score/Python36/spark-2.3/TradingCustomerChurn/TradingChurnRiskClassificationSparkML/1'}

# Write the test data without label to a .csv so that we can later use it for batch scoring
write_score_CSV=test.toPandas().drop(['CHURNRISK'], axis=1)
write_score_CSV.to_csv('../datasets/TradingCustomerSparkMLBatchScore.csv', sep=',', index=False)

# Write the test data to a .csv so that we can later use it for Evaluation
write_eval_CSV=test.toPandas()
write_eval_CSV.to_csv('../datasets/TradingCustomerSparkMLEval.csv', sep=',', index=False)
```



Data
Scientist

The model is saved in the CPD repository, which can provide governance to the data and the model itself.

7.5 Real-time and batch scoring

Now that the model is created, you will score and evaluate it.

- _298. Click project [TradingCustomerChurn](#) ⇒ [Models](#)

Click on the model to open it.

The screenshot shows the IBM Watson Studio interface. At the top, there is a breadcrumb navigation: Home > Projects > TradingCustomerChurn. A green checkmark is placed above the 'Projects' link. Below the navigation, the project name 'TradingCustomerChurn' is displayed in large blue text. Underneath the project name, there are several tabs: Assets (17), Data Sources (2), Jobs (0), Environments (1), and Collaborators (1). The 'Assets' tab is currently selected. To the left, there is a sidebar titled 'Recent' with options: Data sets (11), Notebooks (1), Scripts (0), Models (1), Model groups (0), and Analytics dashboards (4). A green checkmark is placed next to the 'Models' option. On the right, there is a search bar labeled 'Search by model name' and a 'Models' section. The 'Models' section shows one entry: 'TradingChurnRiskClassificationSparkML v1'. A green checkmark is placed next to this entry. The entire interface is enclosed in a light gray border.

- _299. The first test on this data will be a real-time score of the model using sample input values.

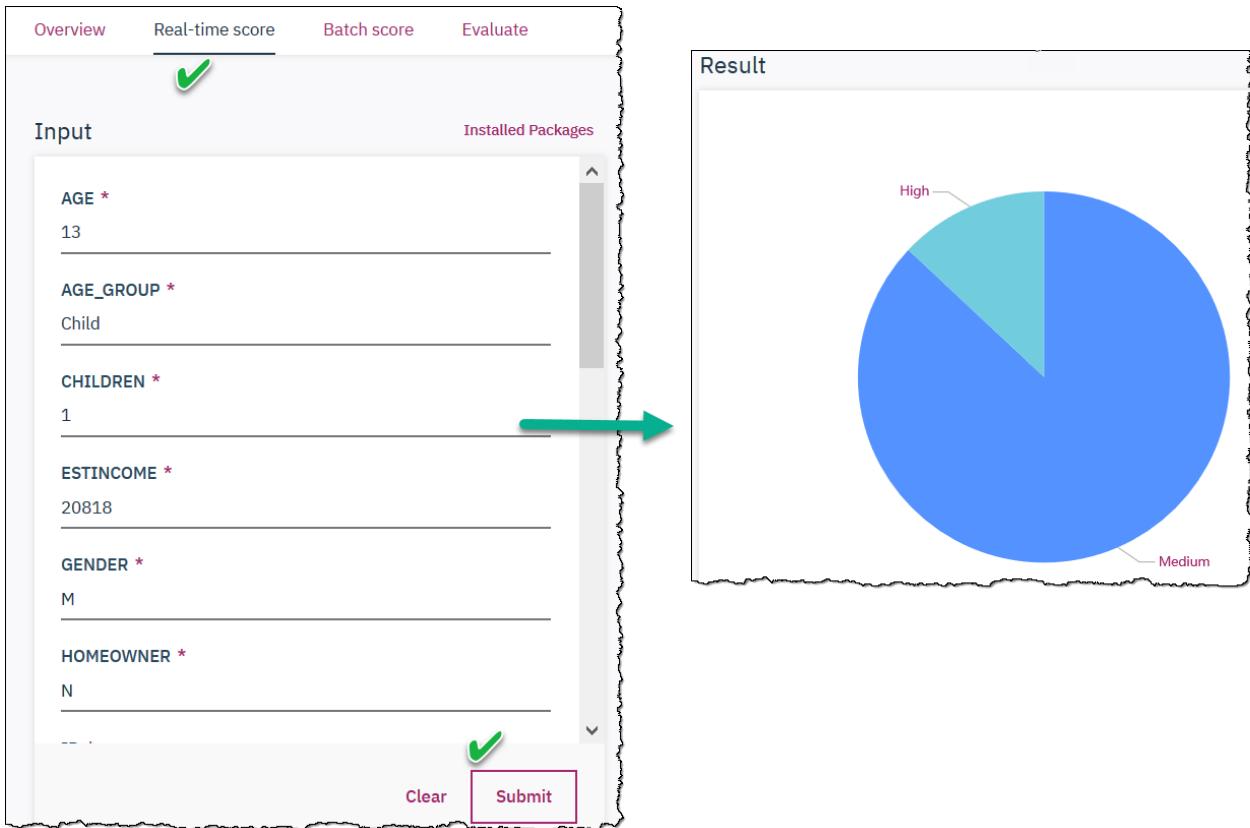
To do this, click [Real-time score](#)

The screenshot shows the 'TradingChurnRiskClassificationSparkML' model page. At the top, there is a breadcrumb navigation: TradingCustomerChurn > Models > TradingChurnRiskClassificationSparkML. A green checkmark is placed above the 'Models' link. Below the navigation, there are four tabs: Overview, Real-time score (which is highlighted with a green checkmark), Batch score, and Evaluate. The 'Real-time score' tab is currently active. The page content is mostly blank at this stage.

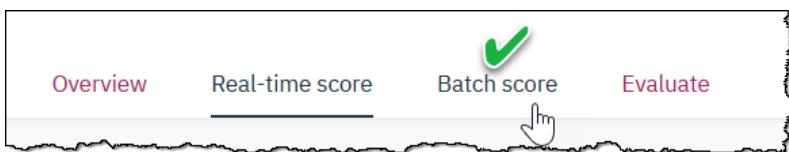
- _300. Notice the sample input values can be changed - use what is provided by default for this first score.

The screenshot shows an 'Input' form. At the top, there is a button labeled 'Install'. Below the button, there is a field for 'AGE *' with the value '13' entered. Below this, there is a field for 'AGE_GROUP *' with the value 'Child' entered. The entire form is enclosed in a light gray border.

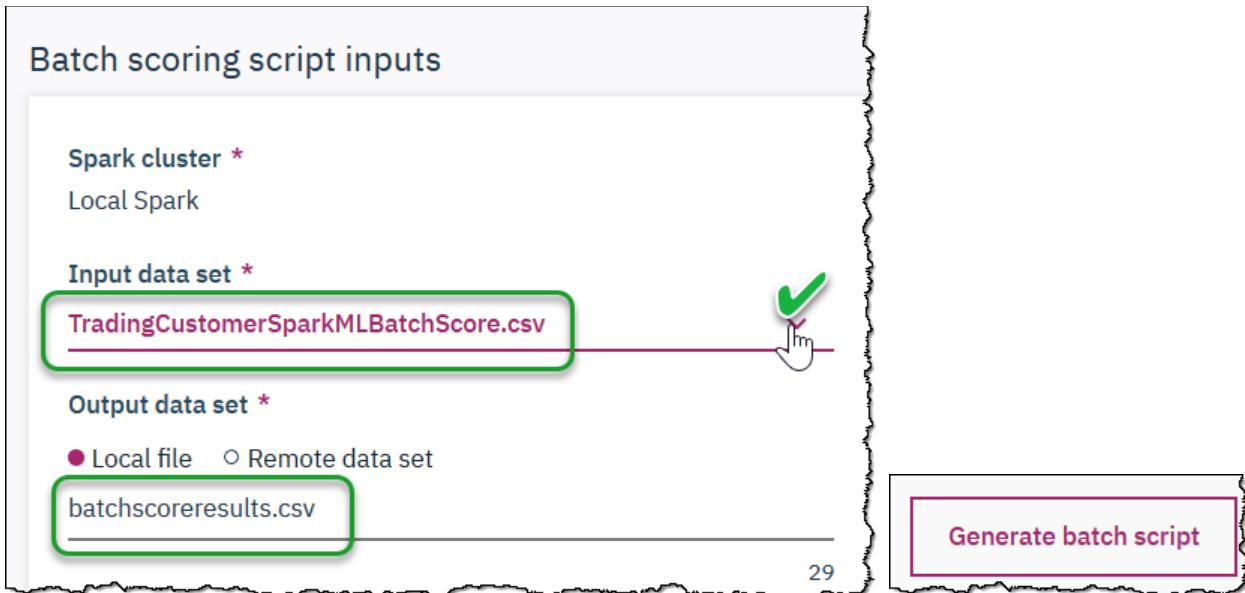
__301. Click **Submit**. The model Real-time scoring shows the percentage of High vs. Medium likelihood of churn for this particular set of input values. (Your results may vary if your default Input values are different.)



__302. Click **Batch Score**.



__303. Select Input data set: [TradingCustomerSparkMLBatchScore.csv](#) and type the output file name [batchscoreresults.csv](#). Click [Generate batch script](#).



29

__304. Click [Run now](#).

Result

```

1 #!/usr/bin/python
2
3 import sys, os
4 from pyspark.sql import SparkSession
5 from pyspark.ml import Pipeline, Model, PipelineModel
6 from pyspark.sql import SQLContext
7 import pandas
8 import dsx_core_utils, re, jaydebeapi
9 from sqlalchemy import *
10 from sqlalchemy.types import String, Boolean
11
12
13 # setup dsxr environmental vars from command line input
14 from dsx_ml.ml import dsxr_setup_environment
15 dsxr_setup_environment()
16
17 # define variables
18 args = {'execution_type': 'DSX', 'target': '/datasets/batchscoreresults.csv', 'source': '/datasets/TradingCustomerSparkMLBatchScore.csv', 'output_type': 'CSV'}
19 input_data = os.getenv("DEF_DSX_DATASOURCE_INPUT_FILE", (os.getenv("DSX_PROJECT_DIR") + args.get("source")))
20 output_data = os.getenv("DEF_DSX_DATASOURCE_OUTPUT_FILE", (os.getenv("DSX_PROJECT_DIR") + args.get("target")))
21 model_path = os.getenv("DSX_PROJECT_DIR") + os.path.join("/models", os.getenv("DSX_MODEL_NAME", "TradingChurnRiskClassificationSparkML"), os.getenv("DSX_MODEL_VERSION"))
22
23 # create spark context
24 spark = SparkSession.builder.getOrCreate()
25 sc = spark.sparkContext
26
27 # read test dataframe (inputJson = "input.json")
28 testDF = SQLContext(sc).read.csv(input_data, header='true', inferSchema='true')
29
30 # load model
31 model_rf = PipelineModel.load(model_path)
32
33 <
34

```

Run now

__305. Scroll down to **Runs** and click on the **ID**.

ID	NAME	TARGET HOST	TRIGGERED BY	STARTED AT	DURATION (S)	RESULT
1559101369-999	Run 1	Local instance	admin	28 May 2019, 10:42 PM	In progress	Pending

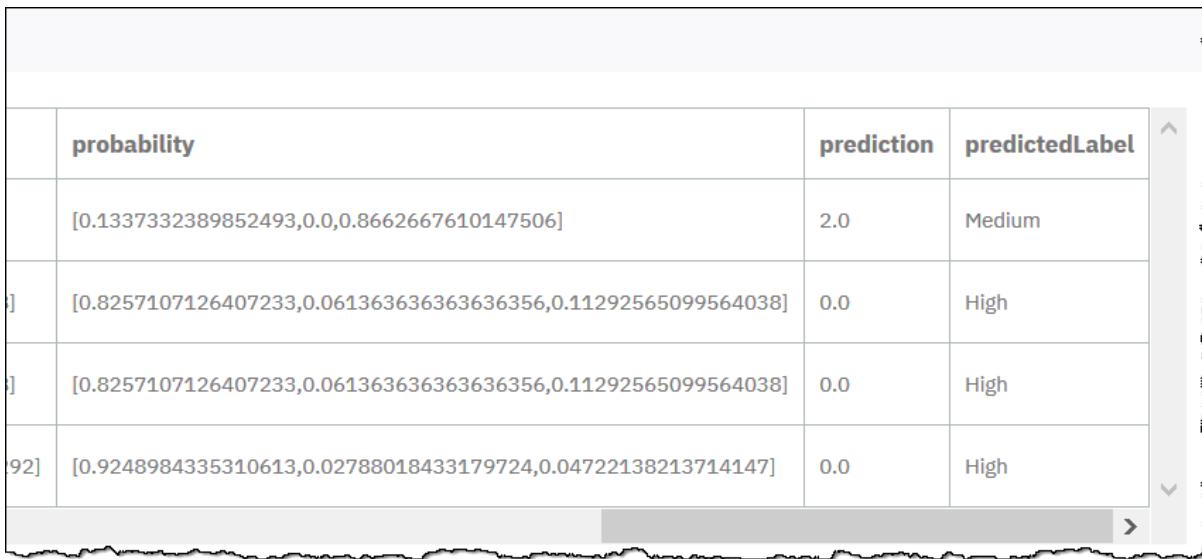
__306. When the **DURATION** has a number, it means the job is finished.

__307. Return to the Project **TradingCustomerChurn** \Rightarrow **Assets** \Rightarrow **Data sets**

Find **batchscoreresults.csv**

__308. Preview this dataset

- __309. Scroll to the right and check the prediction for the sample data for different input data combinations.

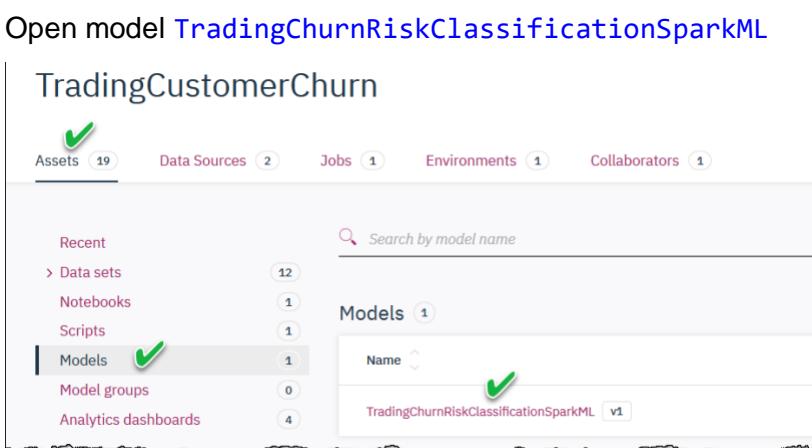


	probability	prediction	predictedLabel
	[0.1337332389852493,0.0,0.8662667610147506]	2.0	Medium
]]	[0.8257107126407233,0.0613636363636356,0.11292565099564038]	0.0	High
]]	[0.8257107126407233,0.0613636363636356,0.11292565099564038]	0.0	High
92]	[0.9248984335310613,0.02788018433179724,0.04722138213714147]	0.0	High

- __310. [Close](#) the batch results preview

7.6 Create an Evaluation

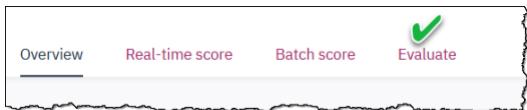
- __311. Return to the Project [TradingCustomerChurn](#) \Rightarrow [Assets](#) \Rightarrow [Models](#)



The screenshot shows the 'Assets' tab selected in the top navigation bar of the IBM Watson Studio interface. Below the navigation bar, there are sections for 'Recent', 'Data sets', 'Notebooks', 'Scripts', 'Models', 'Model groups', and 'Analytics dashboards'. The 'Models' section is expanded, showing a table with one entry: 'TradingChurnRiskClassificationSparkML' (v1). A green checkmark is placed next to the 'Models' link in the sidebar and next to the model entry in the table.

Name	v1
TradingChurnRiskClassificationSparkML	v1

__312. Click **Evaluate**



__313. Select from first drop-down Input data set [TradingCustomerSparkMLEval.csv](#)

Select Evaluator **Multiclass**

Select Threshold metric **F1 score**

Set Threshold criteria to be between **0.71** to **0.89**.

Click [Generate evaluation script](#).

Model evaluation script inputs

Spark cluster *

Local Spark

Input data set *

[TradingCustomerSparkMLEval.csv](#)

Evaluator *

Multiclass

Threshold metric *

F1 Score

Threshold *

0 ————— [Slider] ————— 1
Min: 0.71 Mid: 0.89

Generate evaluation script

__314. The evaluation script is generated and you can run it by clicking [Run now](#).

```

1 #!/usr/bin/python
2
3 import pandas as pd
4 import json
5 from uuid import uuid4
6 import time, sys, os, shutil, glob, io, requests
7 from pyspark.sql import SparkSession
8 from pyspark.ml import Pipeline, Model, PipelineModel
9 from pyspark.sql import SQLContext
10 import dsx_core_utils
11 from dsx_ml.ml import save_evaluation_metrics
12
13
14 # setup dsxr environmental vars from command Line input
15 from dsx_ml.ml import dsxr_setup_environment
16 dsxr_setup_environment()
17
18 # define variables
19 args = {"dataset": "/datasets/TradingCustomerSparkMLEval.csv", "published": "false", "threshold": {"metric": "f1Score", "min_value": 0.71, "mid_value": 0.71, "max_value": 0.71}, "model_path": os.path.join(os.getenv("DSX_PROJECT_DIR"), "models", os.getenv("DEF_DSX_MODEL_NAME"), "TradingChurnRiskClassificationSparkML"), os.getenv("DEF_DSX_DATA_SOURCE_INPUT_FILE")}
20
21 # create spark context
22 spark = SparkSession.builder.getOrCreate()
23 sc = spark.sparkContext
24
25 # Load the input data
26
27 input_data = os.getenv("DEF_DSX_DATA_SOURCE_INPUT_FILE", os.getenv("DSX_PROJECT_DIR") + args.get("dataset"))
28 dataframe = SQLContext(sc).read.csv(input_data, header="true", inferSchema="true")
29
30 # Load the model from disk
31 model_rf = PipelineModel.load(model_path)
32
33 <
34

```

__315. The evaluation job is scheduled and at first is [Pending](#) \Rightarrow [Running](#)

Click on the [Id](#)

[DURATION](#) shows the job completed

Runs						
Id	Name	Target Host	Triggered By	Started At	Duration (S)	Result
1559059066-999	Run 1	Local instance	admin	28 May 2019, 10:57 AM	In progress	Pending

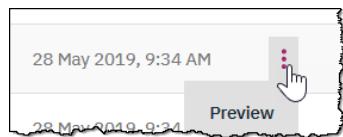
DURATION
48 s

__316. Return to the Project [TradingCustomerChurn](#) ⇒ [Assets](#) ⇒ [Data sets](#)

Find [TradingCustomerSparkMLEval](#)

The screenshot shows the 'TradingCustomerChurn' project page. At the top, there are tabs for Assets (20), Data Sources (2), Jobs (2), Environments (1), and Collaborators (1). The 'Assets' tab is selected and highlighted with a green checkmark. Below this, there's a 'Recent' section with a dropdown menu for 'Data sets' (also highlighted with a green checkmark) which lists CSV (5), TABLE (7), Notebooks (1), Scripts (2), Models (1), Model groups (0), and Analytics dashboards (4). To the right, there's a search bar labeled 'Search by data set name'. Underneath, a table titled 'Data sets (12)' shows a list of datasets with columns for Name, Type, and Size. The first dataset is 'batchscoreresults.csv' (CSV, 120.0). The second dataset, 'TradingCustomerSparkMLEval.csv', is highlighted with a green checkmark.

⇒ Ellipsis ⇒ Preview



__317. Notice one of the high risk for churn candidates in this evaluation matches the same criteria of what the business analyst found in the heatmap of his dashboard.

Preview - TradingCustomerSparkMLEval.csv

The screenshot shows a preview of the 'TradingCustomerSparkMLEval.csv' file. The table has 12 columns: AGE, AGE_GROUP, CHILDREN, CHURNRISK, ESTINCOME, GENDER, HOMEOWNER, ID, STATUS, DAYSSINCELASTLOGIN, and DAYSSINCELASTTRADE. There are 5 rows of data. The fourth row, which corresponds to an ID of 1715, is highlighted with a green box. This row has an AGE of 14, an AGE_GROUP of Child, CHILDREN of 2, a CHURNRISK of High, an ESTINCOME of 86695, a GENDER of F, a HOMEOWNER of N, an ID of 1715, a STATUS of M, and both DAYSSINCELASTLOGIN and DAYSSINCELASTTRADE of 7.

AGE	AGE_GROUP	CHILDREN	CHURNRISK	ESTINCOME	GENDER	HOMEOWNER	ID	STATUS	DAYSSINCELASTLOGIN	DAYSSINCELASTTRADE
13	Child	1	Medium	20818	M	N	1044	M	4	13
14	Child	0	High	15924	M	N	1631	S	0	0
14	Child	0	High	15924	M	N	1754	S	0	0
14	Child	2	High	86695	F	N	1715	M	7	7
14	Child	2	Low	86695	F	N	1310	M	1	9



Data Scientist

Models must be periodically tested, trained and redeployed. This is where you will see even more value of CPD as an integrated data platform.

__318. Check out results under [Models](#) ⇒ [Overview](#) ⇒ [Evaluation results](#)

The screenshot shows the CPD interface. At the top, there is a navigation bar with 'TradingCustomerChurn' > 'Models' > 'TradingChurnRiskClassificationSparkML'. A green checkmark is placed over the 'Models' link. Below this, the model name 'TradingChurnRiskClassificationSparkML v1' is displayed, with a subtitle 'This is a SparkML Model to Classify Trading Customer Churn Risk'. A green checkmark is placed over the model name. A horizontal navigation bar below the title includes 'LAST MODIFIED' (24 Jun 2019, 9:20 AM), 'TYPE' (Spark), and buttons for 'Overview' (highlighted with a green checkmark and a red arrow pointing to it), 'Real-time score', 'Batch score', and 'Evaluate'. In the 'Evaluation results' section, a table displays performance metrics: START TIME (24 Jun 2019, 9:30 AM), ACCURACY (0.95), F1 SCORE (0.94), WEIGHTED PRECISION (0.95), WEIGHTED RECALL (0.95), MIN THRESHOLD (0.71), MID THRESHOLD (0.89), and PERFORMANCE (Good, highlighted with a green checkmark). A green checkmark is also placed over the 'Evaluation results' heading.

7.7 Lab conclusion

You have seen how to create, score and evaluate models. But continued success for your organization will be by creating repeatable processes to prepare, test, evaluate and deploy of the models that have been infused into the applications.

The CPD platform provides the agility required for this.

The next lab exercise will show the deployment and the repeatable process that can fit into your CICD pipe line of DevOps.

**** End of Lab 07: Analyze Part 2 – Model Creation**

Lab 08 Deploy and Infuse

8.1 Lab overview

In our previous lab exercise, we created a model, scored and evaluated it using the CPD console. Business value is achieved when a successful model is deployed, infused into an application and subsequently updated. In this [Deploy and Infuse](#) lab you will learn how CPD assists you in this regard.

In the case of Boatswain Trading, it is their Stock Trader application that will benefit from this particular exercise.

 Developer	<p>We are calling this lab Deploy and Infuse even though you will see some materials on Cloud Pak for Data refer to this stage as Deploy, while others call it Infuse.</p> <p>The thing is, since deployed models allow for the infusion of machine learning into applications, this stage can be described as both.</p>
--	--

8.2 Persona represented in this lab

The [Developer](#) persona is the likely one to perform the various [Deploy and Infuse](#) tasks in this lab. However, the Data Scientist persona could perform the CPD specific tasks as well.

Persona (Role)	Capabilities
 Developer	Developers create and maintain the end-user applications that utilize the output from all the other personas on the CPD platform.

8.3 Deploy the model

Think of model deployment as the equivalent of writing a self-service application that takes the model and makes it available through a REST API interface. Application developers access and consume the model through the same interface. While this is a manual process in most organizations, Cloud Pak for Data can automate deploying and maintain models without writing a single line of code.

CPD eliminates the need for to do the following:

- Write code to perform the above capability and use a runtime to deploy it.
- Create a runtime on bare metal machines that require OS installation, network, storage, etc.
- Create a runtime on a virtual machine in a VMware on Intel, or IBM POWER VM® on a POWER platform.
- Create a runtime in Docker that requires someone to build the image and deploy it on one of the above platforms.

Each of the above requires manpower and machine resources. Using CPD, you can bypass this and quickly harvest insight from your data in a repetitive fashion by integrating it with your CICD pipeline.

8.3.1 Commit changes

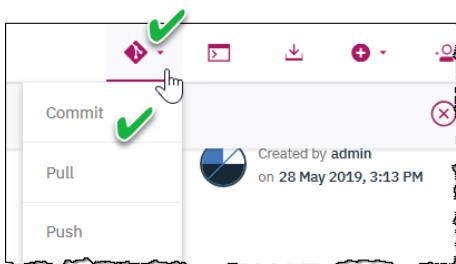
You will now examine the automated delivery and node deployment which you can integrate with a DevOps implementation.

- _319. From the [Navigation menu](#) ⇒ [Projects](#) ⇒ [TradingCustomerChurn](#)

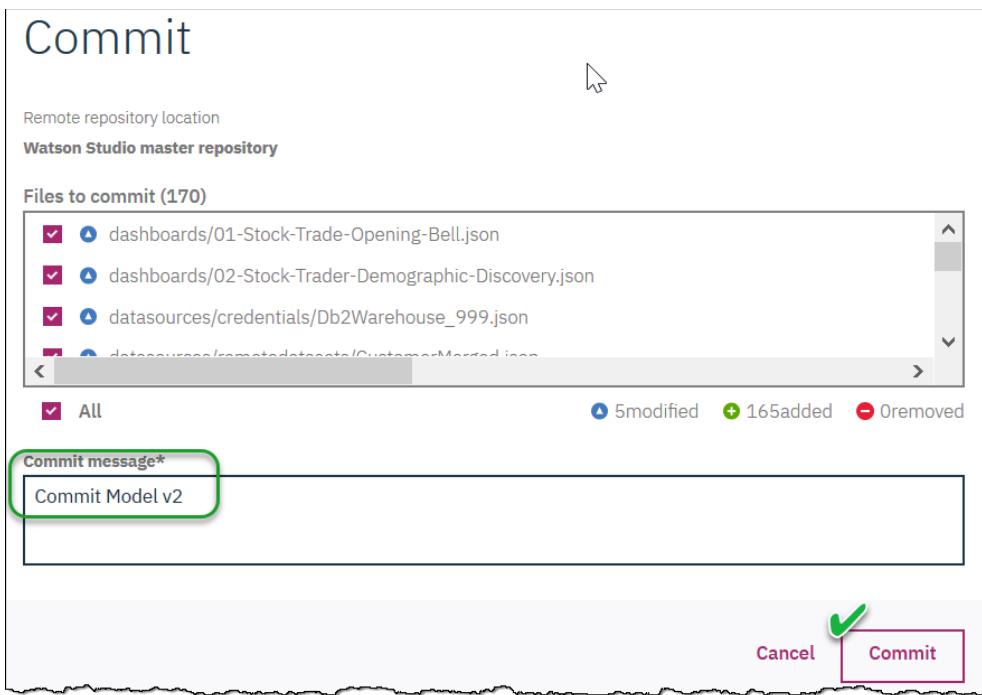
You should see this message because you have made changes to your project.



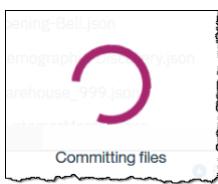
- _320. Click [Git](#) icon and click [Commit](#).



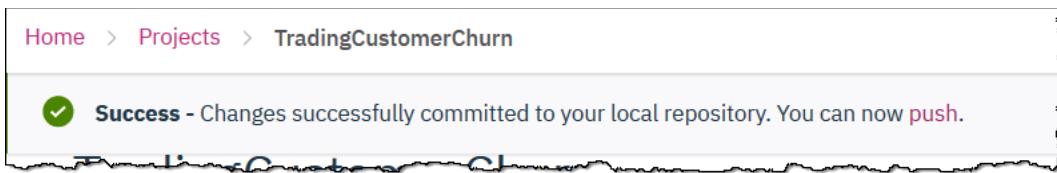
- _321. Type: *Commit message* [Commit Model v2](#) ⇒ [Commit](#).



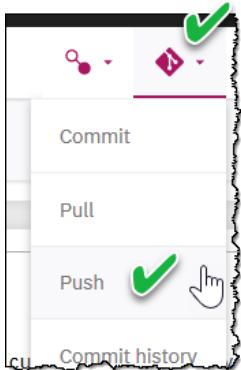
- _322. Wait while the files are committed



__323. When completed you will see the following message.



__324. Now [push](#) your changes.



__325. Use tag [v2model](#) ⇒ [Push](#) (Note: If v2model is already there choose v3model)

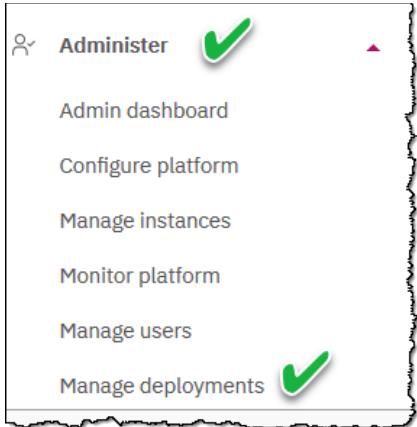


__326. The changes have been pushed to the master repository. [X](#) out to delete this message.



8.3.2 Model management and deployment

_327. Click [Navigation menu](#) ⇒ [Administer](#) ⇒ [Manage deployments](#)



_328. A [Project Release](#) called `stocktrader` has already been created. This was used to deploy the model used in the Executive Demo you ran through earlier.

You will now use this same [Project Release](#) to deploy another version of the model, the one you created in the previous lab.

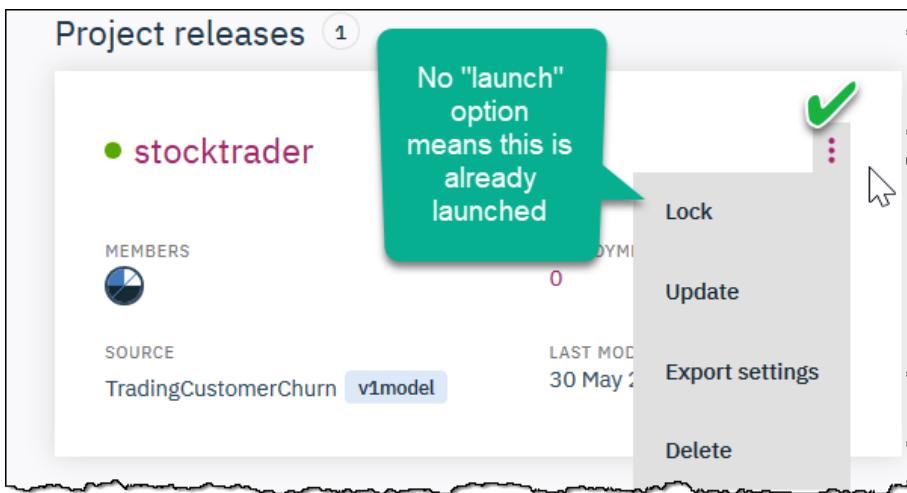
A screenshot of the "Project releases" screen. It shows a single entry for "stocktrader". The details are: MEMBERS (empty), DEPLOYMENTS (0), SOURCE (TradingCustomerChurn v1model), and LAST MODIFIED (30 May 2019, 12:56 PM). There is a three-dot menu icon on the right.



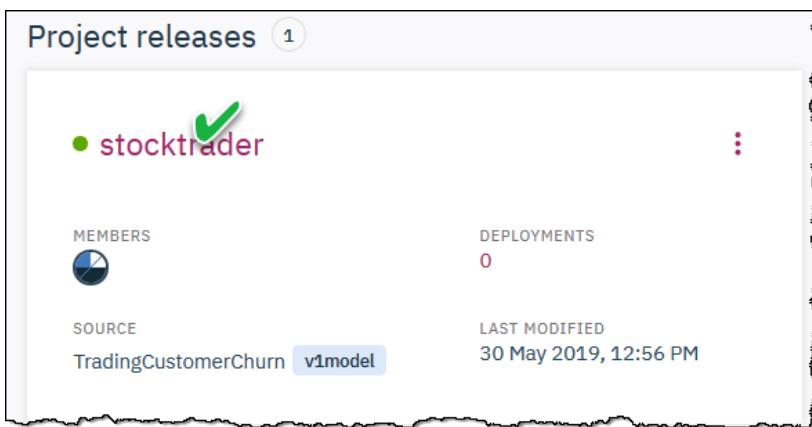
Even though this was done for you, creating a Project Release is relatively simple: Click [+ Add Project Release](#) and fill in the screen with a Name, Route, Source Project, and Tag.

__329. Click the [Ellipsis](#) to view the options for this Project release.

If [Launch](#) is available, then [Launch](#) the Project release. (Note: it should already be launched and look like the image below.)

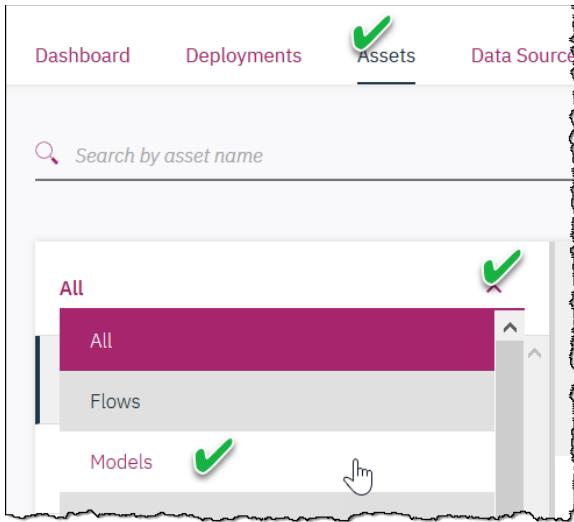


__330. In the Project release tile for [stocktrader](#), click on the actual text “[stocktrader](#)” to open it.



__331. Click tab **Assets**

Filter with drop down and choose **Models**



__332. Find the model with the name: **TradingChurnRiskClassificationSparkML**

A screenshot of a software interface showing a list of models on the left and detailed information for one model on the right. The left sidebar shows a dropdown menu with 'Models' selected, followed by a list of three models: 'TradingChurnRiskClassification...', 'TradingChurnRiskClassification...', and 'TradingChurnRiskClassification...'. A green arrow points from the 'Models' dropdown to the first model in the list. The right side displays the details for the first model: Name: 'TradingChurnRiskClassificationSparkML', Version: 1, Last Modified: '28 May 2019, 10:37 PM', Type: 'spark-2.3', Engine: 'Python36', Algorithm: 'Classification'. Below this, there is a table with columns 'Name', 'Asset', 'Type', and 'Visibility'. The first row shows 'stocktrader' and 'TradingChurnRiskClassificationSparkML v1' under 'Asset', 'Web service' under 'Type', and '-' under 'Visibility'. There is also a green checkmark icon next to the model name 'TradingChurnRiskClassificationSparkML'.

__333. On the right of display of that model, click + web service



_334. Name: stocktrader \Rightarrow Model version: 1 \Rightarrow Web service environment: Python 3.6 – Script as a Service \Rightarrow Reserve CPU cores \Rightarrow Reserve GB Memory \Rightarrow Replicas:1 \Rightarrow Create

Name *
stocktrader

URL
<https://10.77.200.160:31843/dmodel/v1/award/pyscript/stocktrader>

Model version *
1

Web service environment *
Python 3.6 - Script as a Service

Reserve CPU cores ✓
0

Reserve GB Memory ✓
0

Replicas
1 ✓

Create

_335. The Kubernetes pod deployment is being created. This will take a few minutes to complete.

In the meantime, review the scoring Endpoint that can be embedded into your Stock Trader application. The Developer can easily click on the copy icon to paste it into his application code.

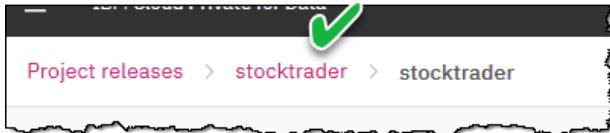


_336. In the same manner, a Deployment Token was created for the Developer to use.



The model deployment through CPD is secured with an authentication token. Any applications using the model need to authenticate themselves by providing this token.

_337. Click [stocktrader](#) from the path bar at the top of your screen to go back to that screen



_338. Click tabs [Deployments](#). It should be [Enabled](#) by this time. If is not, refresh your screen until it is enabled.

NAME	ASSET	TYPE	VISIBILITY	DATE STARTED	AVAILABILITY
stocktrader	TradingChurnRiskClassificationSparkML v2	Web service	—	30 May 2019, 1:07 PM	Enabled

The CPD platform builds a Docker container using the underlying private cloud platform. (In this case it is IBM Cloud Private.) The container and its Kubernetes pod are deployed onto the cluster and once deployed, they are ready to service the API requests to provide results back from the model.

This is what the enabled (Running) pod for this deployment looks like on the Kubernetes cluster:



Developer

```
[root@node1 ibmuser]
[root@node1 ibmuser]# kubectl get pod -n zen | grep award
award-pyscript-stocktrader-9df8548d8-vt7gr          1/1      Running   0
[root@node1 ibmuser]#
```

This is an example of the automation that the CPD platform provides in support of integrating data and application development pipelines.

_339. Click the deployment [stocktrader](#) to enter it again.

NAME	ASSET	TYPE	VISIBILITY	DATE STARTED	AVAILABILITY
stocktrader	TradingChurnRiskClassificationSparkML v2	Web service	—	30 May 2019, 1:07 PM	Enabled

stocktrader

ENDPOINT	POST https://10.77.200.160:31843/dmodel/v1/award/pyscript/stocktrader/scor
TYPE	Web service

8.4 Test the deployment

340. In the stocktrader deployment, click tab API.

Review the Function name and Body, and then click Submit

Review the Response output.

The screenshot shows the 'API' tab selected in the top navigation bar. On the left, under 'Request', the 'Function name' is set to 'score' and the 'Body' contains a JSON input string. A green checkmark icon is placed over the 'Submit' button. On the right, under 'Response', the output is displayed as a JSON object with 'result' and 'probabilities' fields, followed by a 'predictions' field containing 'Low'. Below this, 'stdout' is shown with a table structure for 'AGE', 'AGE_GROUP', 'CHILDREN', 'DAYSSINCETRADING', and 'DAYSSINCELASTLOGON' with values 12, Child, 0, 41, and an empty row respectively.

341. Review Generate code

The screenshot shows the 'Generate code' interface. It displays a 'curl' command with various headers and a JSON payload. The command is intended to be copied and pasted into a terminal to make a POST request to the API endpoint. A 'Close' button is located in the bottom right corner of the code editor area.

```

1 curl -k -X POST \
2   https://10.77.200.160:31843/dmodel/v1/award/pyscript/stocktrader/score \
3   -H 'Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmtZSI6ImFkbWluIiwicGFja2FnZU5hbWUiOiJzdG9ja3RyYV
4   -H 'Cache-Control: no-cache' \
5   -H 'Content-Type: application/json' \
6   -d '{"args": {"input_json": [{"AGE":12,"AGE_GROUP":"Child","CHILDREN":0,"ESTINCOME":28770,"GENDER":"F","HOMEOWNER":"N","ID":1}]}'
    
```

342. Below is an example of running the curl command in a Terminal window.

(You can try this for yourself if you want to. From the desktop open a [Terminal](#) ⇒ Type: [root](#) ⇒ Copy the generated curl command from the CPD console and paste it into the terminal ⇒ [Enter](#).)

8.5 The deployment infused in the application Stock Trader After

The machine learning model was created by data scientists and deployed using CPD platform. You will now see how a modern microservices-based application can consume it.

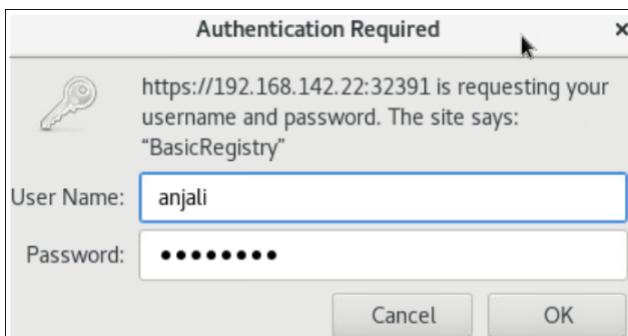
You reviewed the [Stock Trader After](#) application in the Executive Demo lab, but you will now review it in more detail.

__343. Find and double-click desktop icon: [Stock Trader After](#).



344. Log in as *Username*: **anjali** and *Password*: **password**.

Click OK.



__345. Notice the offer box that shows after logging in to the application.

The screenshot shows the 'Summary' page of the IBM TRADER application. At the top, there's a navigation bar with 'Home', 'Summary' (which is highlighted in black), 'Add Portfolio', and 'Predictive Analysis'. Below the navigation bar, there's a 'Change User' button. The main content area has a heading 'Summary' and a sub-headline 'Welcome to IBM Trader powered by ICP for Data'. On the left, there's a list of options with radio buttons: 'Create a new portfolio', 'Retrieve selected portfolio' (which is selected), 'Update selected portfolio (add stock)', and 'Delete selected portfolio'. To the right of this list is a green checkmark icon followed by a box containing text: 'no processing fee for next 5 trades + 1 free wealth management session with certified planner'. Below this box is an 'Advertisement' section featuring a banner for 'IBM Cloud Private for Data'.

Based on user ID of the login, you get the demographics information about that user from the database which then provides that information to a REST API call from the backend of the microservice.

The demographics information is given to the model running on the CPD platform. The model returns a response (the churn prediction value) which is based on the business logic to present a retention offer to the user.

All of this is performed in real time as the user logs into the application.

__346. Click [Change User](#).

The screenshot shows the 'Change User' page of the IBM TRADER application. At the top, there's a navigation bar with 'Home', 'Summary' (highlighted in black), 'Add Portfolio', and 'Predictive Analysis'. Below the navigation bar is a 'Change User' button. There's also a green checkmark icon above the 'Change User' button.

__347. Type User: [dan](#) and Password: [password](#) and see the offer.

The offer is based on the risk assessment of the User ID.

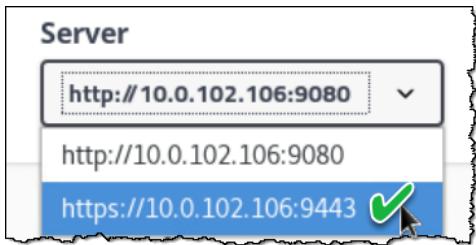
The screenshot shows a user offer message. It features a green checkmark icon and the text 'no processing fee for next 2 trades'.

__348. Let's go further into the REST API call.

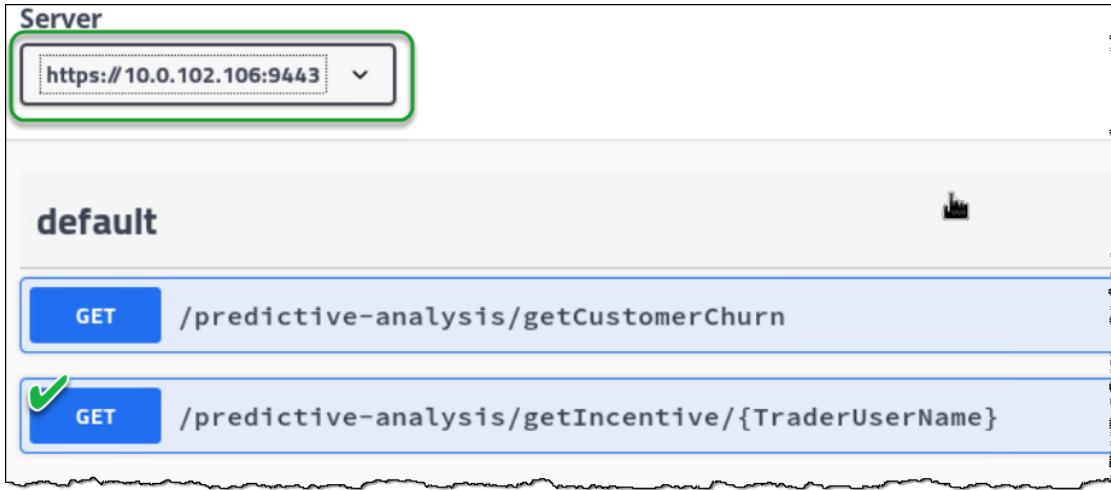
Click [API Explorer](#) desktop icon.



__349. Under **Server**, choose the **https** option.



__350. Click **GET** on the second option: `/predictive-analysis/getIncentive/{TraderuserName}`

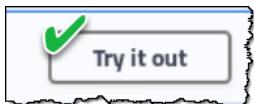


default

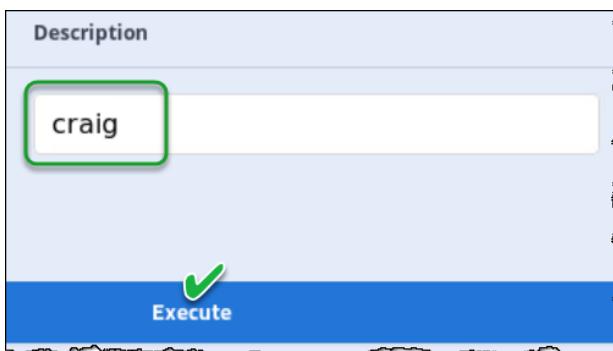
GET /predictive-analysis/getCustomerChurn

✓ GET /predictive-analysis/getIncentive/{TraderUserName}

__351. Click **Try it out**.



__352. Type **craig** and click **Execute**.



Description

craig

✓ Execute

__353. Notice the responses: `curl` command with GET verb, 200 response code, and the response body which is consumed by the application.

The screenshot shows a terminal window with the following content:

```
Code          Details
200 ✓       Response body ✓
{
  "offer": "no processing fee for next 5 trades"
}
```

A green checkmark is placed next to the '200' status code and another green checkmark is placed next to the 'Response body' section.

__354. Try user `foo`. You will notice no response as we have no demographic information for this user.

The screenshot shows a terminal window with the following content:

Name	Description
TraderUserName * required	string (path)
Execute	
Responses	
Curl <code>curl -X GET "https://10.0.102.106:9443/predictive-analysis/getIncentive/foo" -H "accept: application/json"</code>	
Request URL <code>https://10.0.102.106:9443/predictive-analysis/getIncentive/foo</code>	
Server response	
Code	Details
200	Response body
<pre>{ "offer": "" }</pre>	

A large green arrow points from the 'Request URL' section down to the 'Response body' section of the 'Server response' table.

8.6 Lab conclusion

As an organization, we collect and have access to more data than we can possibly analyze. Data is spread all over, in many silos, and is hard to find. Locating that one person in an organization who has the breadth and depth of knowledge of all our data is unlikely.

Cloud Pak for Data is the remedy. It provides an integrated data platform where data engineers and data stewards can prepare the data, data scientists can shop for data and build analytics models, business analysts can provide insights into the data and developers can infuse the results into the applications that run the business.

**** End of Lab 08: Deploy and Infuse**

Lab 09 Wrap-up

9.1 Lab overview

Let's do some wrap up tasks the various personas might do after completing the project.

9.2 Data Scientist wrap-up



Data Scientist

9.2.1 Save the project

The Data Scientist saves her work by [Exporting](#) the project she just completed to a file on the server in a location that the OS administrators consistently backup.

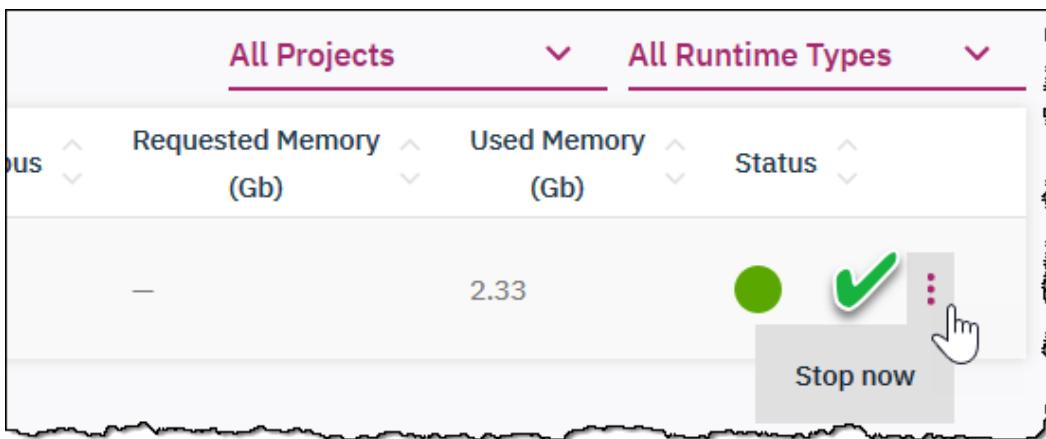
- _355. In the project screen, click on the [Export](#) icon at the top right of the screen

The screenshot shows the IBM Cloud Private for Data interface. At the top, there is a navigation bar with 'IBM Cloud Private for Data' and a search bar. Below the navigation bar, the project 'TradingCustomerChurn' is selected. The main area displays the project name 'TradingCustomerChurn'. Below the project name, there are tabs for 'Assets (14)', 'Data Sources (2)', 'Jobs (0)', 'Environments (1)', and 'Collaborators (1)'. On the far right of the top bar, there is an 'Export' icon (a downward arrow with a checkmark). A green arrow points from this icon to the 'Export' button in a modal dialog. The dialog has the title 'Export project'. It contains sections for 'Export format' (radio buttons for '.zip' and '.tar.gz', with '.zip' selected), 'Current project state' (a dropdown menu), and 'Cancel' and 'Export' buttons. The 'Export' button is highlighted with a pink box and a green checkmark. A second green arrow points from the 'Save File' radio button in the dialog to the 'OK' button, which is also highlighted with a pink box and a green checkmark. The 'OK' button is part of a larger modal dialog titled 'Opening TradingCustomerChurn.zip' which shows the file details: 'TradingCustomerChurn.zip' (Zip archive, 90.2 MB, from https://192.168.142.21:31843) and options to 'Open with Archive Manager (default)' or 'Save File' (selected). There is also a checkbox for 'Do this automatically for files like this from now on.'

9.2.2 Stop the environment

She also saves resources on the CPD cluster by making sure to stop her environment after she is done with it.

- __356. Click [Navigation Menu](#) ⇒ [My Instances](#) ⇒ [Environments](#)
- __357. She clicks on the [ellipsis](#) on the Environment ⇒ [Stop now](#)



9.3 Data Steward wrap-up

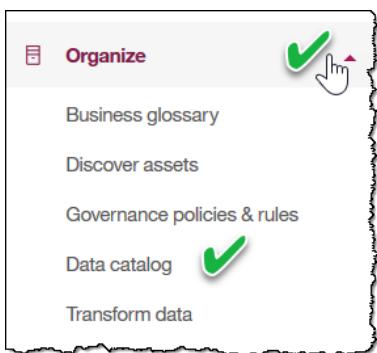


Data Steward

9.3.1 Review the Data catalog

The Data Steward works to maintain the Data catalog to keep it accurate and up-to-date.

- __358. From [Navigation Menu](#) ⇒ [Organize](#) ⇒ [Data catalog](#)



__359. From here, he can use the various Data Catalog features:

Data Exploration: to help him find data in the catalog.

Hierarchies: to display various asset groups in a hierarchical tree that helps him understand the meaning of the asset and its relationship with other assets.

Queries: helps him find and list assets, their properties and their relationships.

Collections: group together assets related to a specific subject.

9.4 Data Engineer wrap-up

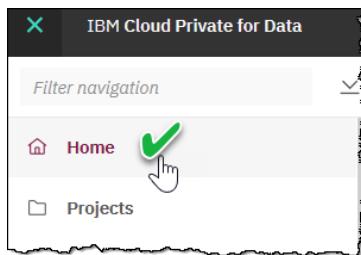


Data Engineer

9.4.1 Publish virtualized tables and views to the catalog

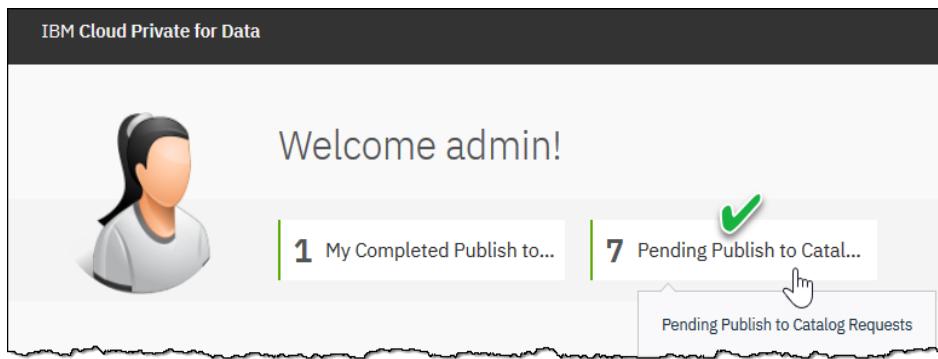
The Data Engineer wants to make the virtualized views she created available to any who wish to shop for them.

__360. From Navigation Menu \Rightarrow Home



__361. After reviewing her home page first, she then...

Selects Pending Publish to Catalog Requests



__362. Selects ellipsis next to the virtualized object she wishes to publish \Rightarrow Approve

Name	Type	Project	Owner	Date Updated	Status	
> USER999.DEMOCHURN_ACTIVITY_VIRTUALIZED	view	-	admin	22 May 2019, 9:22AM	Pending	
> USER999.ALL_ACTIVITY_VIRTUALIZED	table	-	admin	22 May 2019, 9:02AM	Pending	
> USER999.DEMOCHURN_VIRTUALIZED	table	-	admin	22 May 2019, 8:42AM	Pending	

Approve

Reject

9.5 Administrator wrap-up



Administrator

9.5.1 Grant Data Virtualization Access

The Administrator has been asked to open up Data Virtualization capabilities to a trusted user called `ibmuser` which allows the user to create them himself.

- __363. The [Navigation Menu](#) ⇒ [Collect](#) ⇒ [Virtualized data](#)
 - __364. Click [Menu](#)

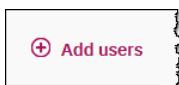


365. Click Manage users

Data virtualization

 | Data sources

- ### 366. Click + Add users



367. Select user `ibmuser` ⇒ [Add](#)

Grant access to users

	Name	Username	Role
<input checked="" type="checkbox"/>	admin	admin	Admin
<input checked="" type="checkbox"/>	ibmuser	ibmuser	User



9.6 Workshop conclusion

With this workshop, you can now see how Cloud Pak for Data turns your organization's data into a critical corporate asset with end-to-end data integration and collaboration on a modern, cloud-native platform.

You have now completed your IBM Journey to Cloud and AI: Analytics Modernization using Cloud Pak for Data.

**** End of Lab 09: Wrap-up**

Appendix A. Stock Trader Opening Bell Dashboard

In this lab, you will build the Stock Trader Opening Bell dashboard from scratch.

We begin by analyzing current trends of customer visits and daily trades in our Stock Trader application. We requested data engineers to provide a file with historical totals of visits and trades for the past year. This file was provided and deposited in our project where we all collaborate.

Build the dashboard

Let's build a dashboard with this data to see if we see any trends.

- __368. From the top left Navigation Menu dropdown click [Projects](#).
- __369. Click project [TradingCustomerChurn](#).
- __370. Under [Assets](#) click [Data Sets](#) \Rightarrow [CSV](#)

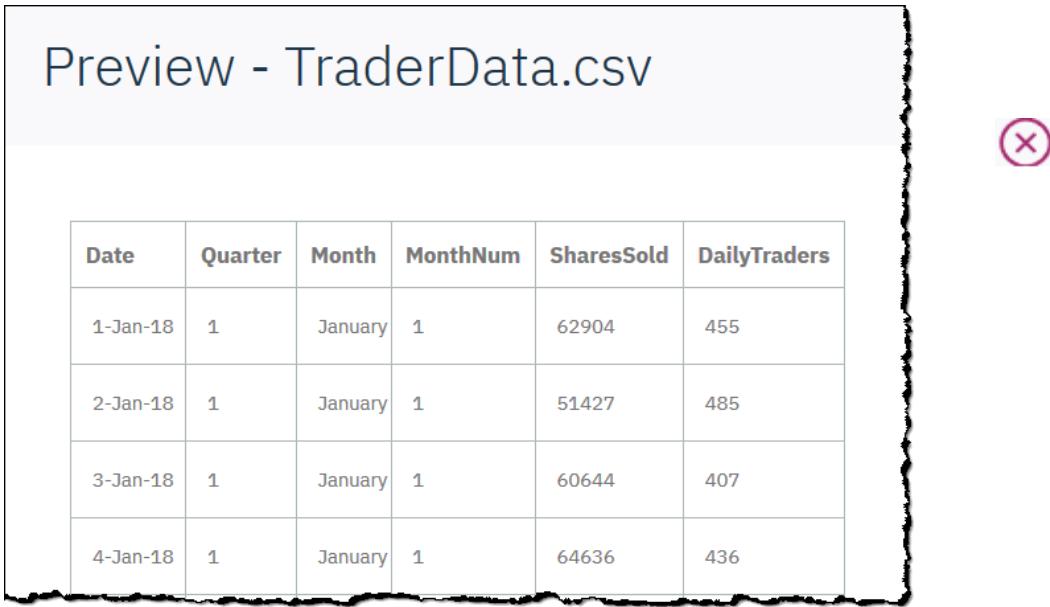
Notice [TraderData.csv](#) available in our project. We will use this data file to build the [Stock Trader – Opening Bell](#) dashboard.

The screenshot shows the Project navigation bar with 'Assets' selected. Below, the 'Data sets' section lists various types of datasets: CSV, TABLE, Notebooks, Scripts, Models, Model groups, and Analytics dashboards. A search bar is present above the list. Two CSV files are listed: 'TraderDataFinal.csv' and 'TraderData.csv'. The 'TraderData.csv' entry is highlighted with a green checkmark.

- __371. Click the ellipsis next to [TraderData.csv](#) then [Preview](#)

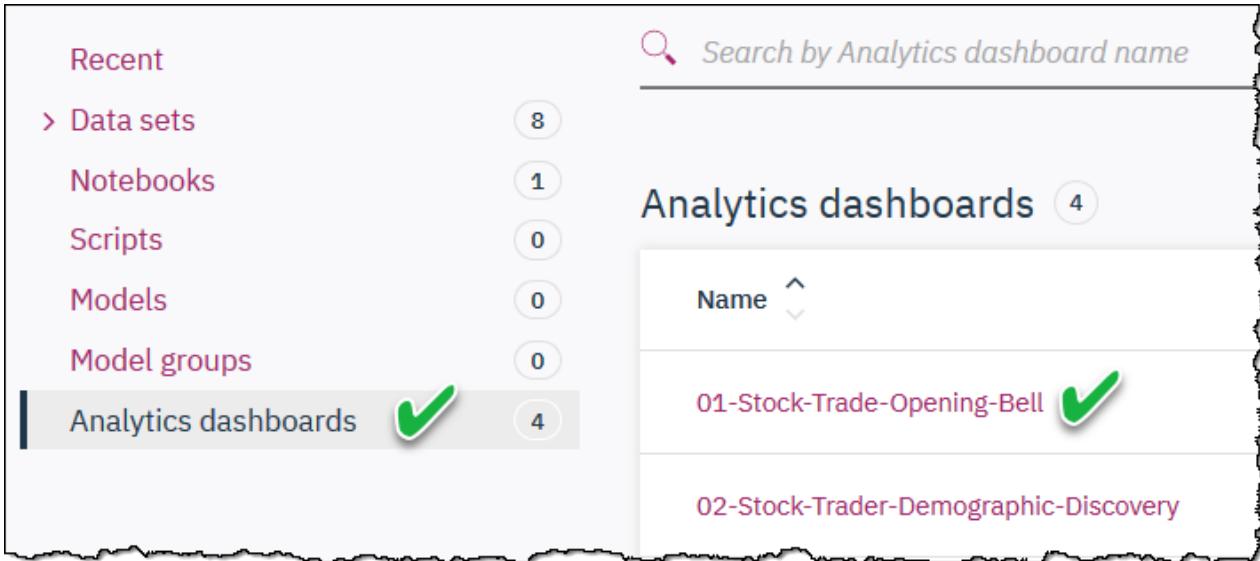
The screenshot shows the 'Data sets' table with two entries: 'TraderDataFinal.csv' and 'TraderData.csv'. A context menu is open over the 'TraderData.csv' row, with the 'Preview' option highlighted.

__372. Peruse the data, then the X to Close  this window.



Date	Quarter	Month	MonthNum	SharesSold	DailyTraders
1-Jan-18	1	January	1	62904	455
2-Jan-18	1	January	1	51427	485
3-Jan-18	1	January	1	60644	407
4-Jan-18	1	January	1	64636	436

__373. Click Analytics dashboards.



Recent

Analytics dashboards 4

Name ↑

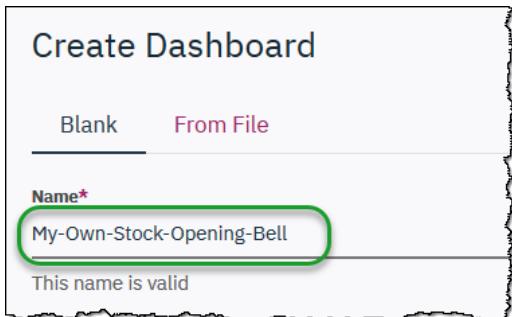
- 01-Stock-Trade-Opening-Bell ✓
- 02-Stock-Trader-Demographic-Discovery
- 03-Stock-Trader-Performance-Analysis
- 04-Stock-Trader-Geographic-Insights

We used the [01-Stock-Trade-Opening-Bell](#) dashboard in the Executive Demo lab and we will build this exact same dashboard here.

__374. Click + Add Analytics Dashboard.



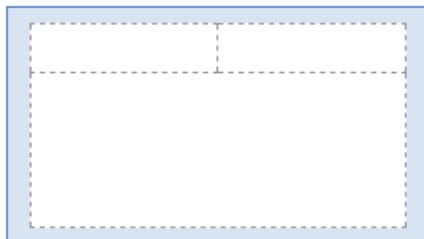
__375. In the [Blank](#) tab, type the Name: [My-Own-Stock-Opening-Bell](#)



__376. Click [Create](#).



__377. You are now presented with a choice of canvas templates. Select the one that looks as shown:



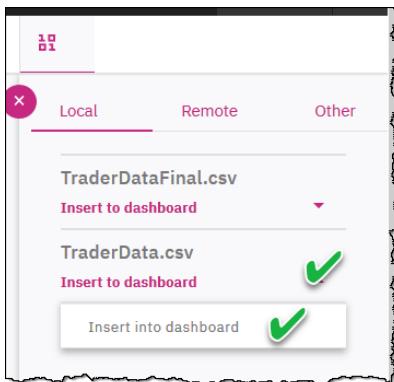
__378. Click [OK](#).

__379. From the [Selected sources](#) area near the top left of your screen, click



__380. The data pane will slide into the right side of the canvas and open to the [Local](#) tab by default.

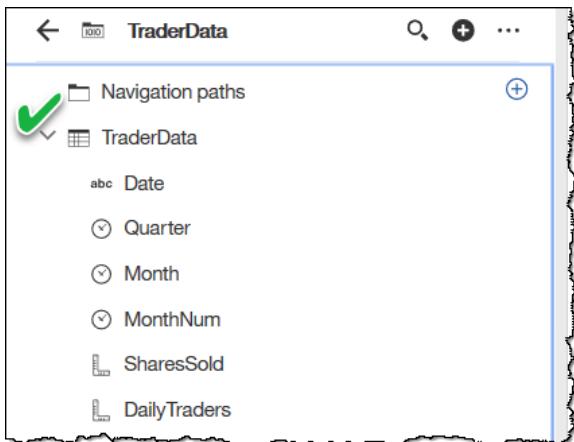
Click the down arrow of [TraderData.csv](#) and click [Insert into dashboard](#).



__381. You will now see [TraderData](#) in the [Selected sources](#) area on the left of the screen.

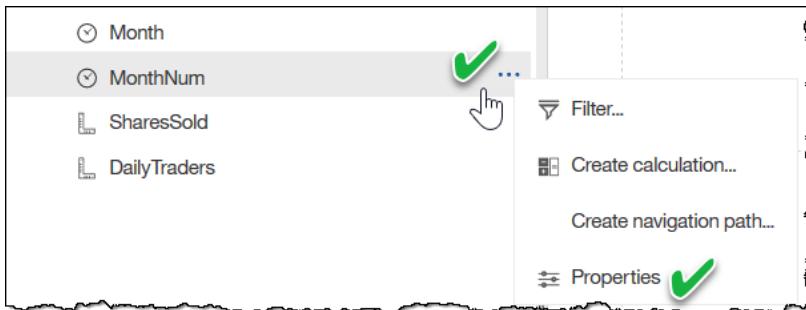


__382. Click [TraderData](#) and expand it to show all the data items in that data source file.

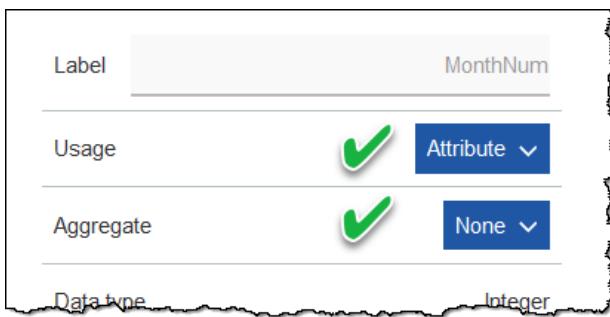


__383. Next we need to make sure the properties of the column data is what we want to be represented correctly within our dashboard.

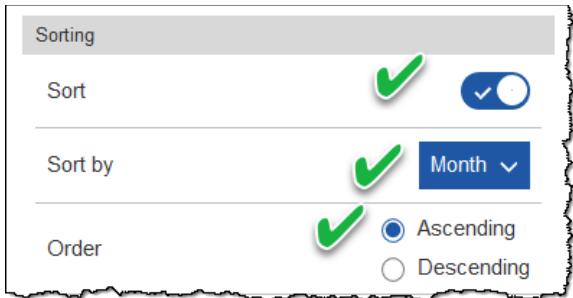
Click the [MonthNum](#) ellipsis, then select [Properties](#) from the flyout menu.



__384. [MonthNum](#) is an attribute on which we sort our months. Change [MonthNum](#) usage to be an [Attribute](#) and Aggregate to be [None](#), then click [Close](#).

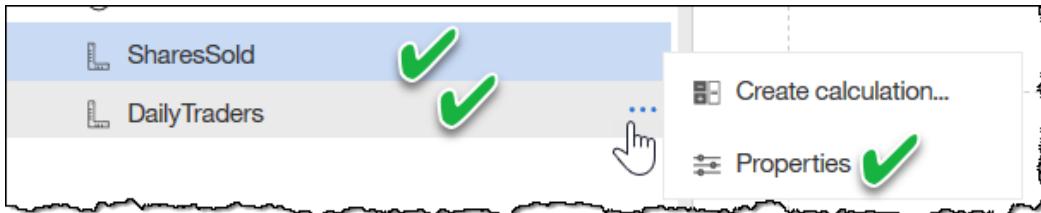


- __385. Select the **Properties** of Month by selecting the ellipse next to Month.
- __386. Slide the **Sort** icon to turn it on and sort by MonthNum in **Ascending** order, then click **Close**.

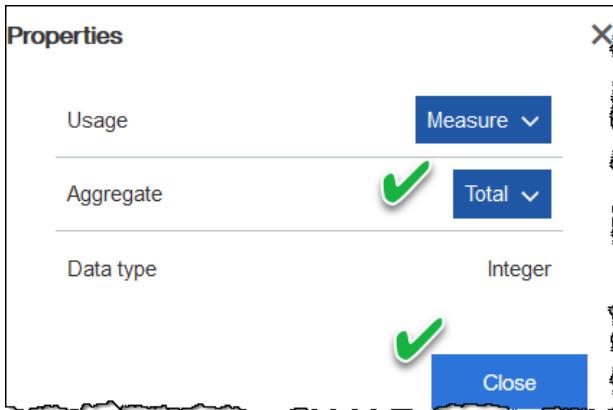


- __387. Next, SET the aggregation of our measures for totaling.

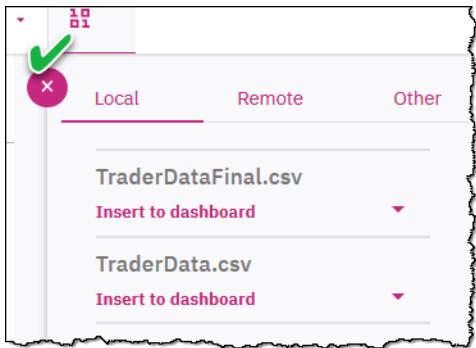
Hold the shift key down and click **SharesSold** and **DailyTraders** to select both.
Click the ellipsis and then click **Properties**.



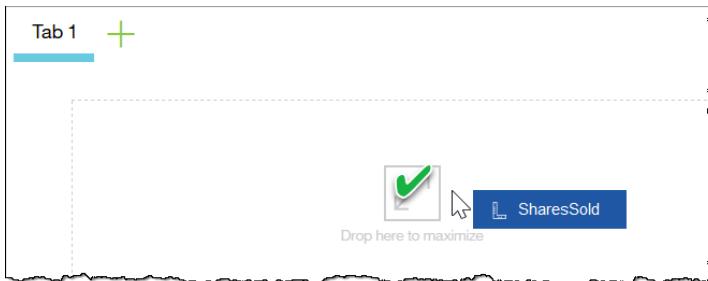
- __388. Change Aggregate to **Total** from the drop-down menu and click **Close**.



- __389. Close the **Data** window at the top right of the screen by clicking the **X**.



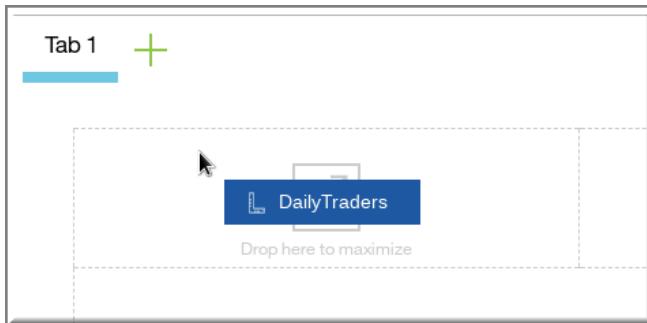
- __390. Drag **SharesSold** to the top left box, hovering over the *Drop here to maximize* area when it turns blue. This gives us a total of **Shares sold** over all time.



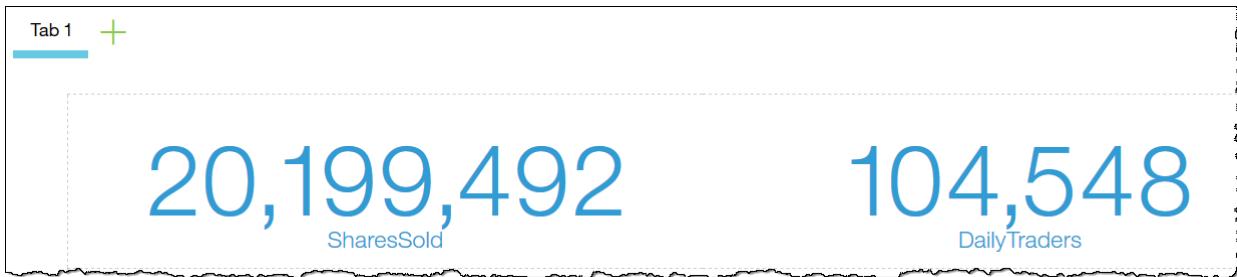
- __391. If you did not drop at the right place it won't fill the template area. You can still adjust the guide to match with the outline of the left box.



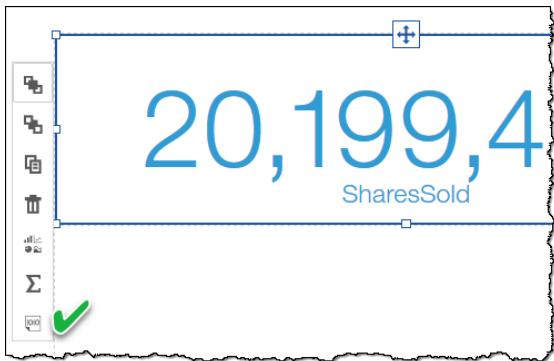
- __392. Drag **DailyTraders** to the top right box, hovering over the *Drop here to maximize* area when it turns blue as well. This maximizes this metric in this box. This gives us a total of trades over all time.



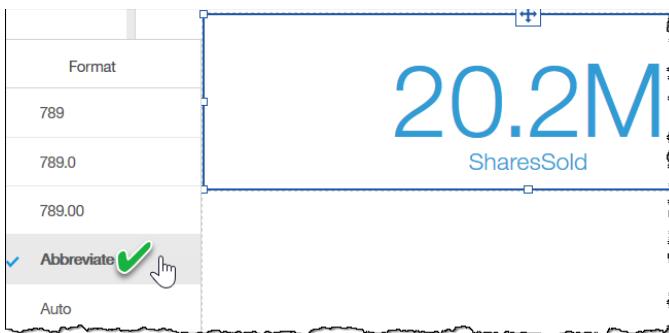
- __393. After you complete both top boxes, the dashboard should display as shown:



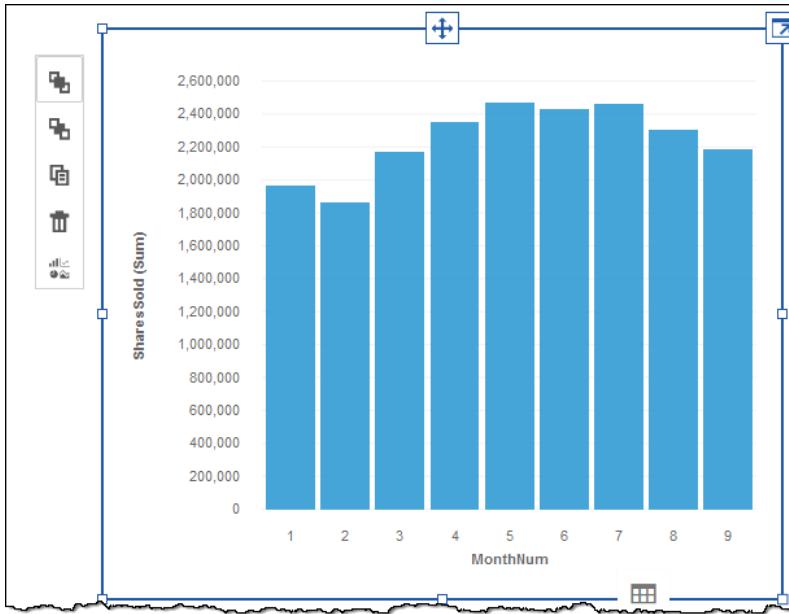
__394. Click the **SharesSold** visualization and select the **format** from the flyout menu.



__395. Click **Abbreviate**. Notice the number in the visualization changes format.



__396. Hold the shift key down, click **Month** and **SharesSold** and drag the two onto the lower canvas area. This time do not drop in the *Drop here to maximize* area.

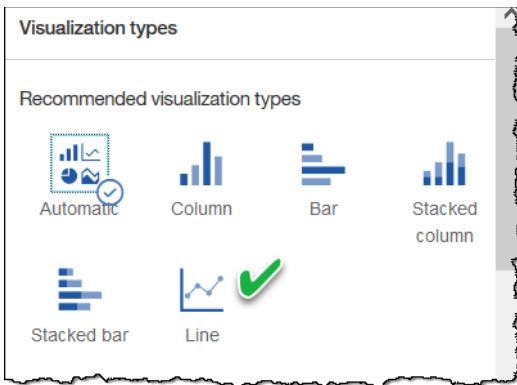


__397. Click the expand icon on top right of the visualization to allow you to make edits to it.

__398. Click the drop-down menu for visualizations which is now set to **Column**.



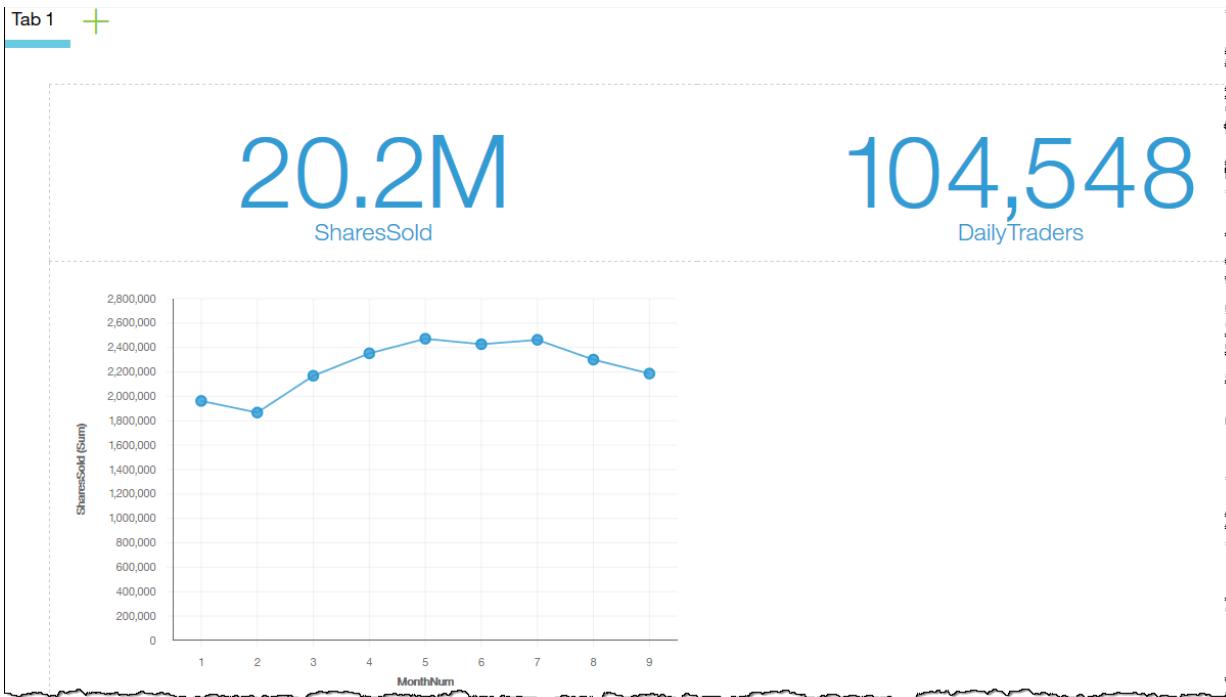
Select **Line** visualization type to change the visualization.



__399. Now select the **Collapse** icon to return the visualization to its original size.



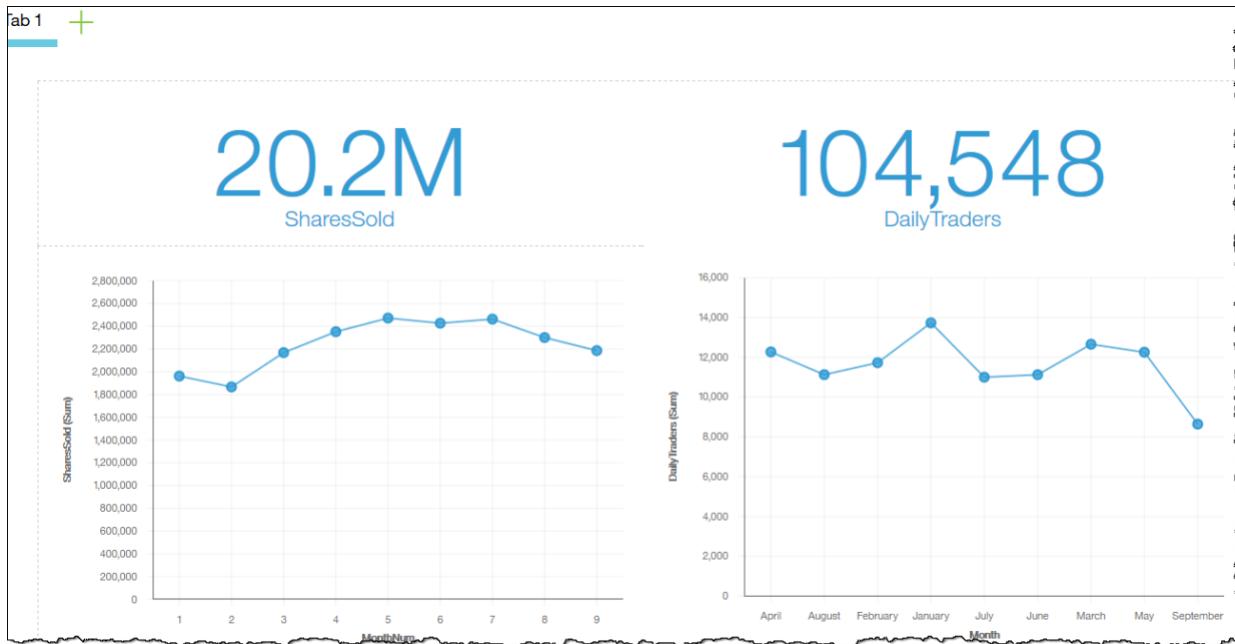
__400. Drag the chart so it aligns with the left side of the work area to look like this:



__401. Hold the shift key down, click **Month** and **DailyTraders** and drag the two onto the lower canvas area. Again do not drop in the *Drop here to maximize* area.

__402. Change the visualization from **Column** to **Line** as we did with the first graph. [Click **Expand**, change the type and click the same button to bring its in original size.]

- __403. Adjust the top, bottom, left and right of the chart boundaries so that all the boxes are aligned.
 __404. The dashboard should display as below.



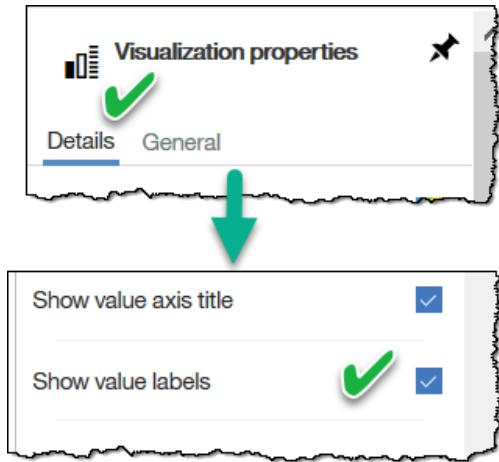
- __405. Click the **Save** icon at the top of the screen to save your work



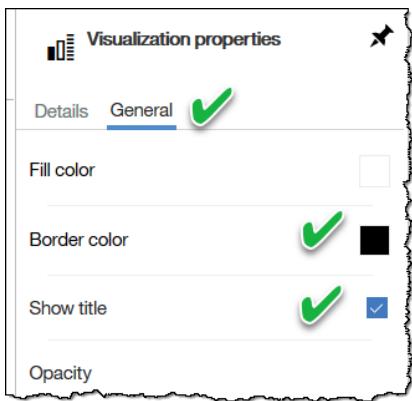
- __406. Select (by clicking on) the bottom left line chart visualization, then select the **Properties** button at the top of the screen to format that visualization.



- __407. From the **Details** tab, check **Show Value Labels**.

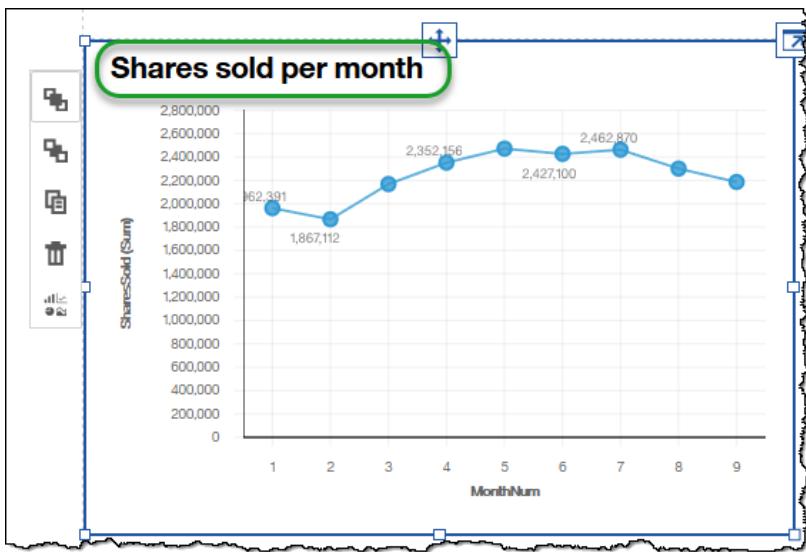


— 408. Now select the **General** tab and check to **Show title** and change **Border color** to black

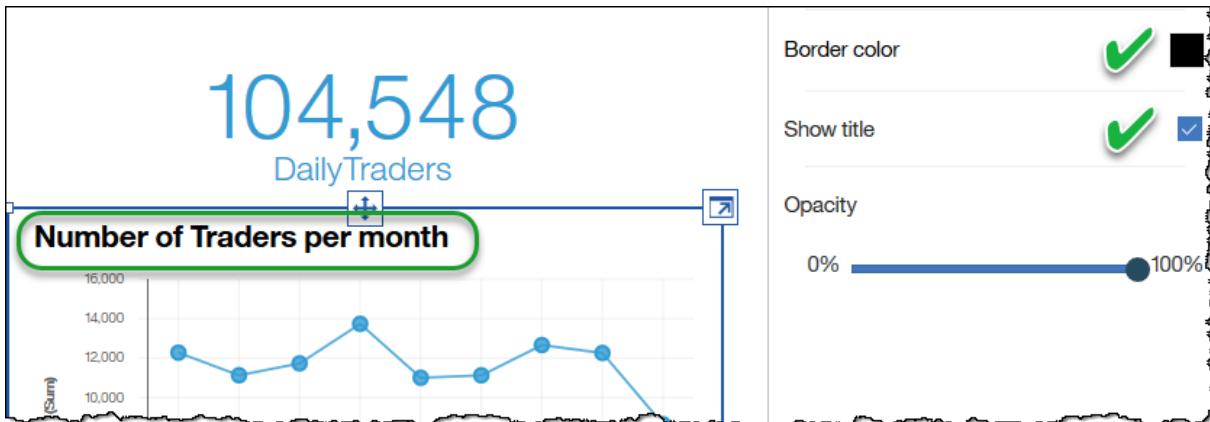


— 409. In the bottom left visualization itself you can now type in a title for it.

Enter **Shares sold per month**.

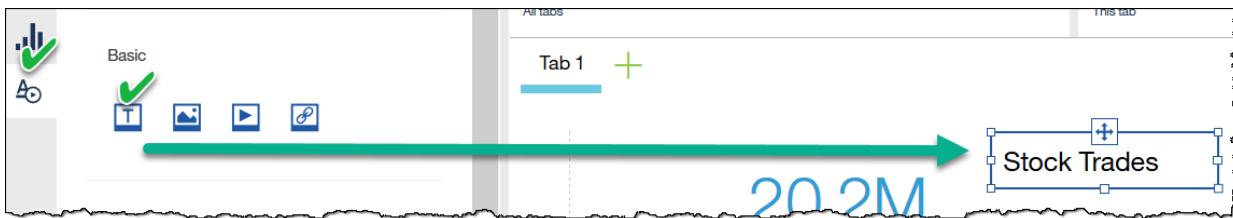


— 410. Click the bottom right visualization and enter **Number of Traders per month** for the **Show title** option and change the **Border color** to black.



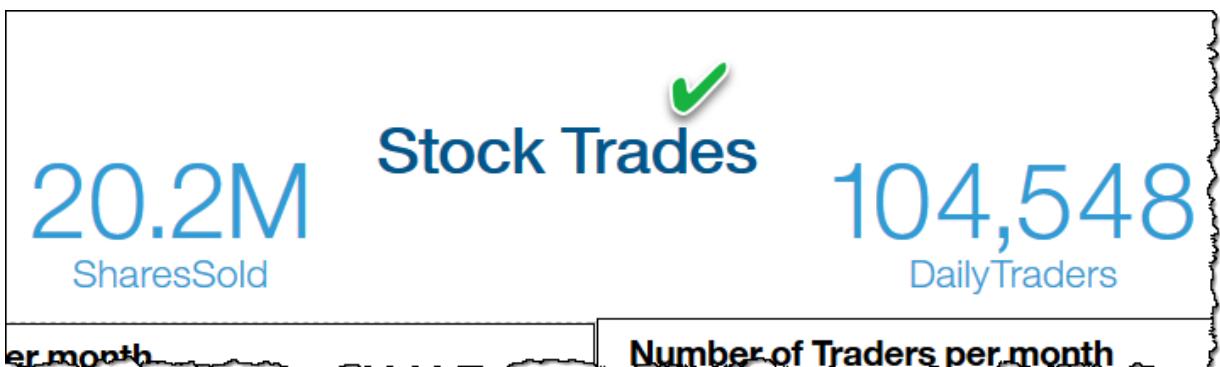
- __411. On the left side of the canvas, click the **Widgets** menu and drag a **Text** box between the top two charts.

Title it **Stock Trades**. Adjust the box so the title stands out.



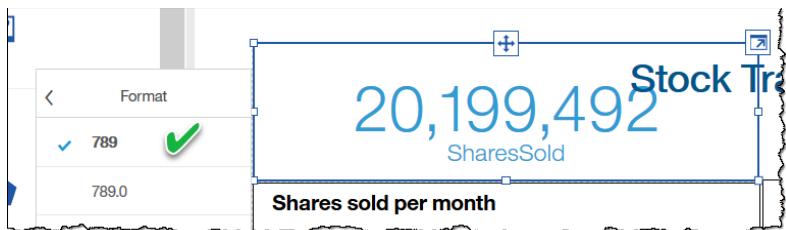
- __412. Format the properties of the title **Stock Trades**. Make the text font size 46, bold, and dark blue in col. Then center it in the text box.

- __413. The final results should look something like this:

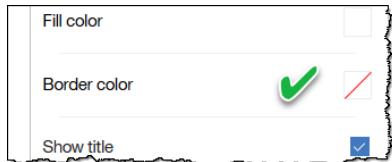


- __414. Click **Save** again to mark your progress here.

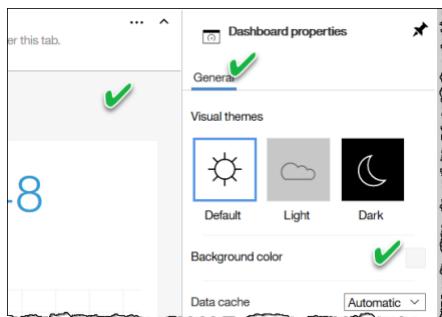
_415. Change the first visualization back from abbreviation to full integer number.



_416. Remove the borders from the bottom visualization by changing the color to nothing.



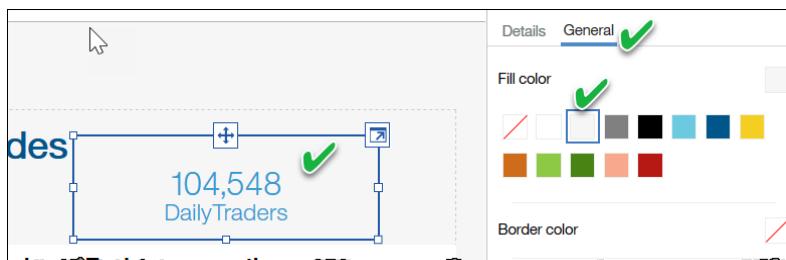
_417. Make the background of the entire dashboard grey by clicking on the work area and not on any visualization. Then from the **Dashboard properties** menu under **General**, change **Background color** to grey.



_418. Adjust the size of the top visualizations to be a bit smaller and centered over each of the bottom visualizations, leaving room for the title to be in full display and not crowded.

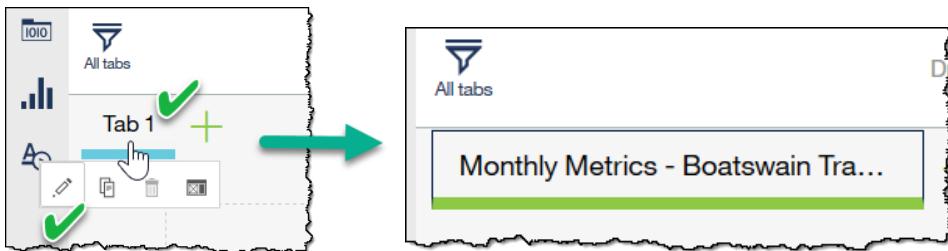


_419. Change the fill color of each of the top visualization to be grey to match the background of the dashboard itself.



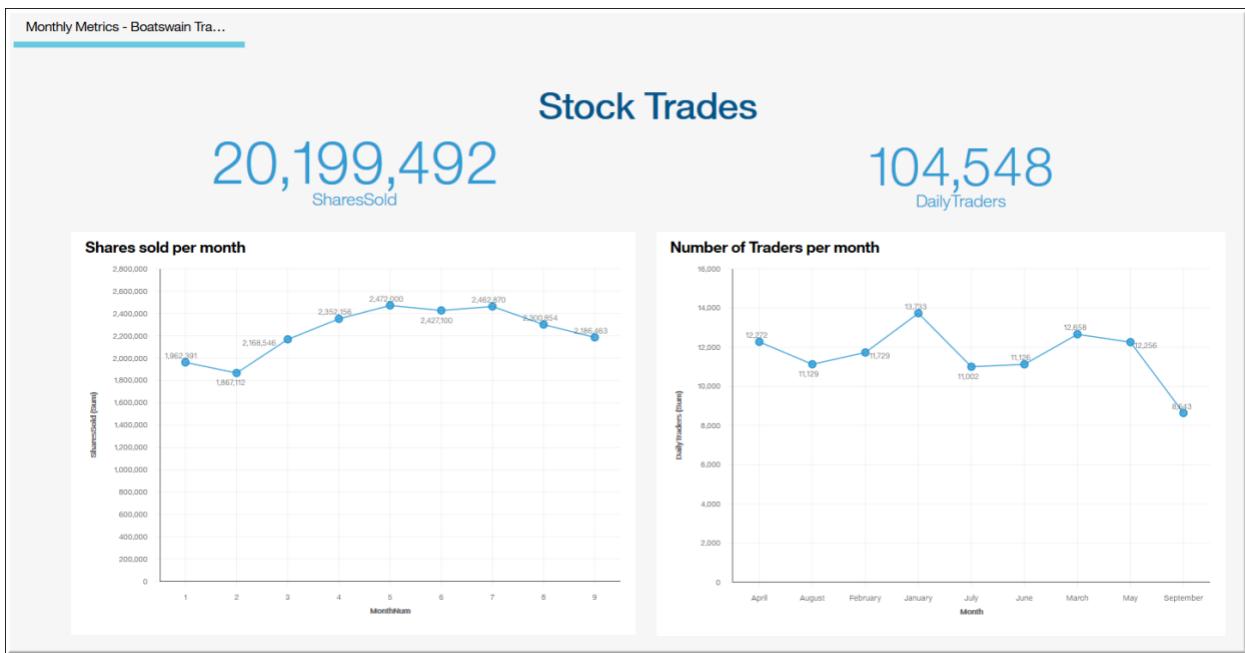
__420. Edit the tab name of this set of visualizations for the dashboard.

Name it: **Monthly Metrics - Boatswain Trading**



__421. Do a final **Save** on this dashboard, then exit the dashboard and come back into it so you will not be in Edit mode.

The final dashboard should look like this:



You will notice from examining the charts that **Shares sold** is relatively flat and daily trades are falling off. We need to use Cloud Pak for Data to discover the WHY behind this trend.

Appendix B. Lab fixes

The following are fixes for lab issues that can come up.

Lab 02 – Executive Demo

9.7.1 Test Connection Fails

1. Test connection can fail if the Db2 user was not created in the pod.

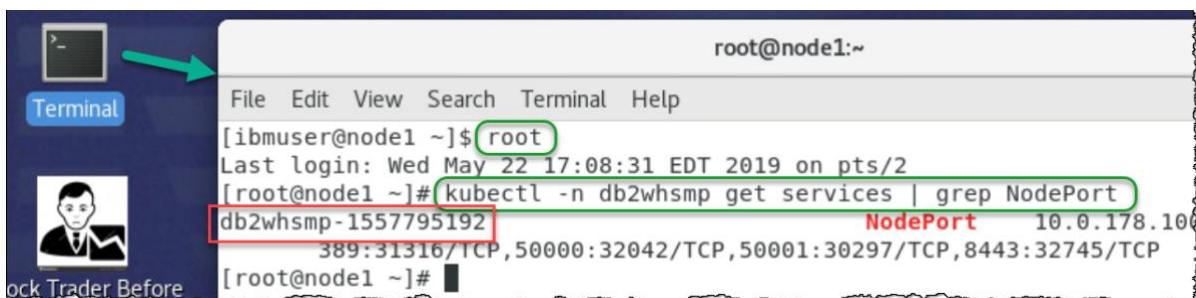
Fix this by clicking on the desktop icon:



2. Test connection can also fail with it not being able to find the host. This means the host service name changed since the last time this workshop cluster was deployed.

You can fix this by doing the following:

- From the Desktop, open a Terminal window
- Type: `root` (this makes you root user)
- Type: `kubectl -n db2whsmp get services | grep NodePort`
- The service name will be something like this: `db2whsmp-nnnnnnnnnn`



- Make sure this service name is the beginning part of the hostname in the JDBC URL string.



- Retry the Test connection again.

9.7.2 Dashboard doesn't render

Sometimes the dashboard doesn't render because the Data Source credentials needs to be re-initialized.

- In the Project go to Data Sources, then find Db2Warehouse.

The screenshot shows the 'Data Sources' section of a project interface. At the top, there are tabs for 'Data Sources' (1), 'Jobs' (0), 'Environments' (1), and 'Collaborators' (1). Below the tabs, there is a search bar labeled 'Search by data source name'. The main area is titled 'Data Sources (1)' and contains a table with columns 'Name', 'Type', and 'Last Modified'. A single row is visible, representing the 'Db2Warehouse' data source. The 'Name' column shows 'Db2Warehouse' with a green checkmark icon above it. The 'Type' column shows 'Db2' and the 'Last Modified' column shows '28 May 2019, 4:13 PM'.

- Retype the Username/Password as below

The screenshot shows a login form with two fields: 'Username *' containing 'icpd' and 'Password *' containing 'icpd'. Below the password field is a checkbox labeled 'Shared'.

- For each data set in this screen, click on it and browse to find the table name in the schema (as shown below.) This will refresh the schema/table for this screen.

The screenshot shows a table titled 'Remote data sets' with columns 'NAME', 'DESCRIPTION', 'TABLE', and 'SCHEMA'. There are two entries:

NAME	DESCRIPTION	TABLE	SCHEMA
CustomerMerged		MERGED_DEMOGRAPHICS_TRADING_CUSTOMER	ICPD
CustomerOfferAccepted		CUSTOMER_OFFER_ACCEPTED	ICPD

The screenshot shows a configuration dialog for a schema. It has a 'Schema' field containing 'ICPD' and a 'Table *' field containing 'MERGED_DEMOGRAPHICS_TRADING_CUSTOMER'. At the bottom right, there is a 'Browse' button with a green checkmark icon above it.

- Click Save for each data set you do this for.
- Click Save for the Data Source itself to save all your work.
- Try the dashboard again.

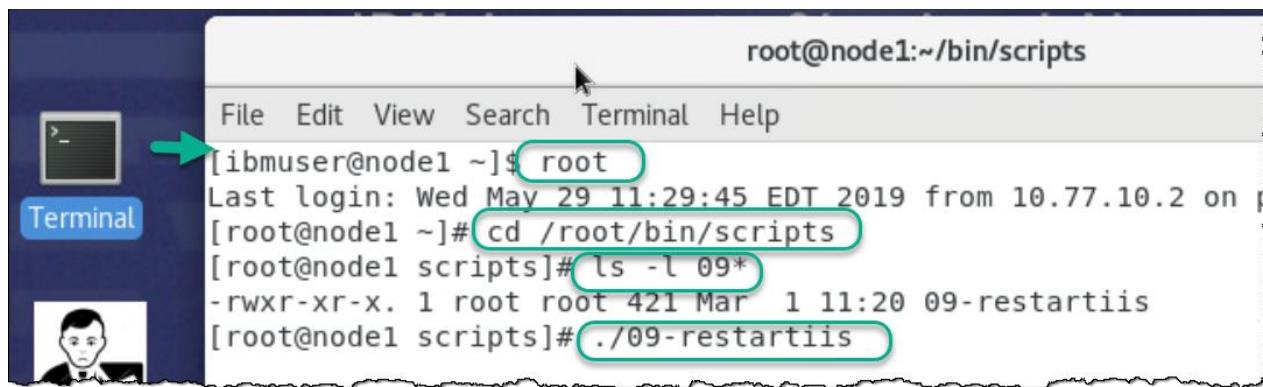
Lab 04 – Organize

9.7.3 Runaway Discover

If the Discover process doesn't complete after 6 or 7 minutes, it may be runaway. You can fix this by doing the following:

- From the Desktop, open a Terminal window, then type:

```
root (this makes you root user)  
cd /root/bin/scripts  
.09-restartiis
```



The screenshot shows a Linux desktop environment with a terminal window open. The terminal window title bar says "root@node1:~/bin/scripts". Inside the terminal, the following command sequence is shown:

```
[ibmuser@node1 ~]$ root  
Last login: Wed May 29 11:29:45 EDT 2019 from 10.77.10.2 on pts/0  
[root@node1 ~]# cd /root/bin/scripts  
[root@node1 scripts]# ls -l 09*  
-rwxr-xr-x. 1 root root 421 Mar 1 11:20 09-restartiis  
[root@node1 scripts]# ./09-restartiis
```

- After this script completes, wait 2 minutes for everything to initialize again in the pod.
- Try the Discover process again.

**** End of Appendix:**

Appendix C. Notices

This information was developed for products and services offered in the USA.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
USA

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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 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.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental. All references to fictitious companies or individuals are used for illustration purposes only.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Appendix D. Trademarks and copyrights

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM	AIX	CICS	ClearCase	ClearQuest	Cloudscape
Cube Views	Db2	developerWorks	DRDA	IMS	IMS/ESA
Informix	Lotus	Lotus Workflow	MQSeries	OmniFind	
Rational	Redbooks	Red Brick	RequisitePro	System i	
System z	Tivoli	WebSphere	Workplace	System p	

Adobe, Acrobat, Portable Document Format (PDF), and PostScript are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, other countries, or both.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both. See Java Guidelines

Microsoft, Windows, Windows NT, and the Windows logo are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

ITIL is a registered trademark and a registered community trademark of the Office of Government Commerce and is registered in the U.S. Patent and Trademark Office.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

Other company, product and service names may be trademarks or service marks of others.



© Copyright IBM Corporation 2019.

The information contained in these materials is provided for informational purposes only and is provided AS IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, these materials. Nothing contained in these materials is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software. References in these materials to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. This information is based on current IBM product plans and strategy, which are subject to change by IBM without notice. Product release dates and/or capabilities referenced in these materials may change at any time at IBM's sole discretion based on market opportunities or other factors and are not intended to be a commitment to future product or feature availability in any way.

IBM, the IBM logo and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at:

<https://www.ibm.com/legal/us/en/copytrade.shtml>

Other company, product and service names may be trademarks or service marks of others.



Please Recycle

