Watson Studio Desktop and Watson Machine Learning Server

Steve Groves, European Team, Data Science and IA
steve.groves@uk.ibm.com

Table of contents

Contents

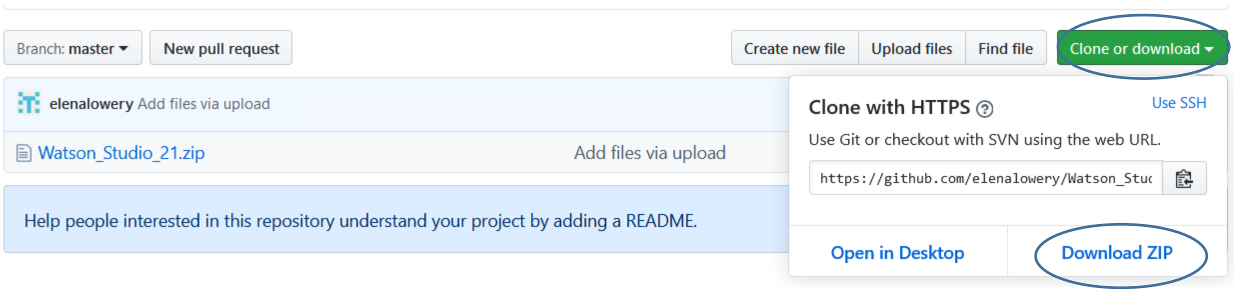
Overview

In this lab you will complete the following tasks in **Watson Studio Desktop (WSD)** and **Watson Machine Learning (WML)** server:

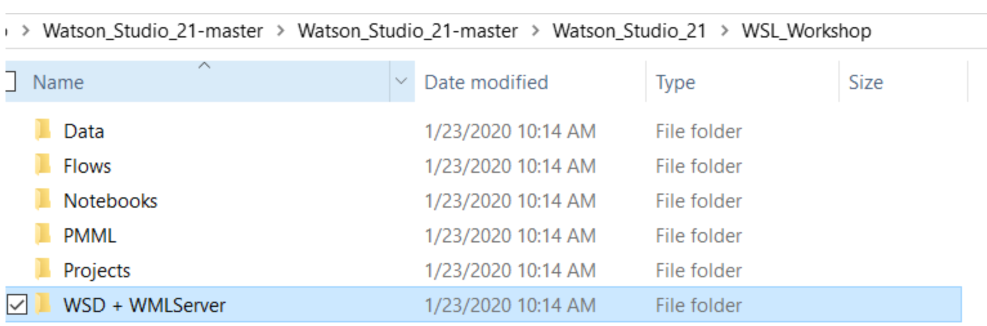
- Create a flow in Watson Studio Desktop.
- Configure and test a connection from WSD to WML Server.
- Access data from a Jupyter Notebook in WSD.
- Deploy models in WML server.

Required software, access, and files

- To complete this lab, you will need:
  - **Watson Studio Desktop 1.0.1** (*perpetual*, not subscription),
  - **Watson Machine Learning Server**
- You will also need to download and unzip this GitHub repository: https://github.com/elenalowery/



- Unzip the files until you get to this directory structure:



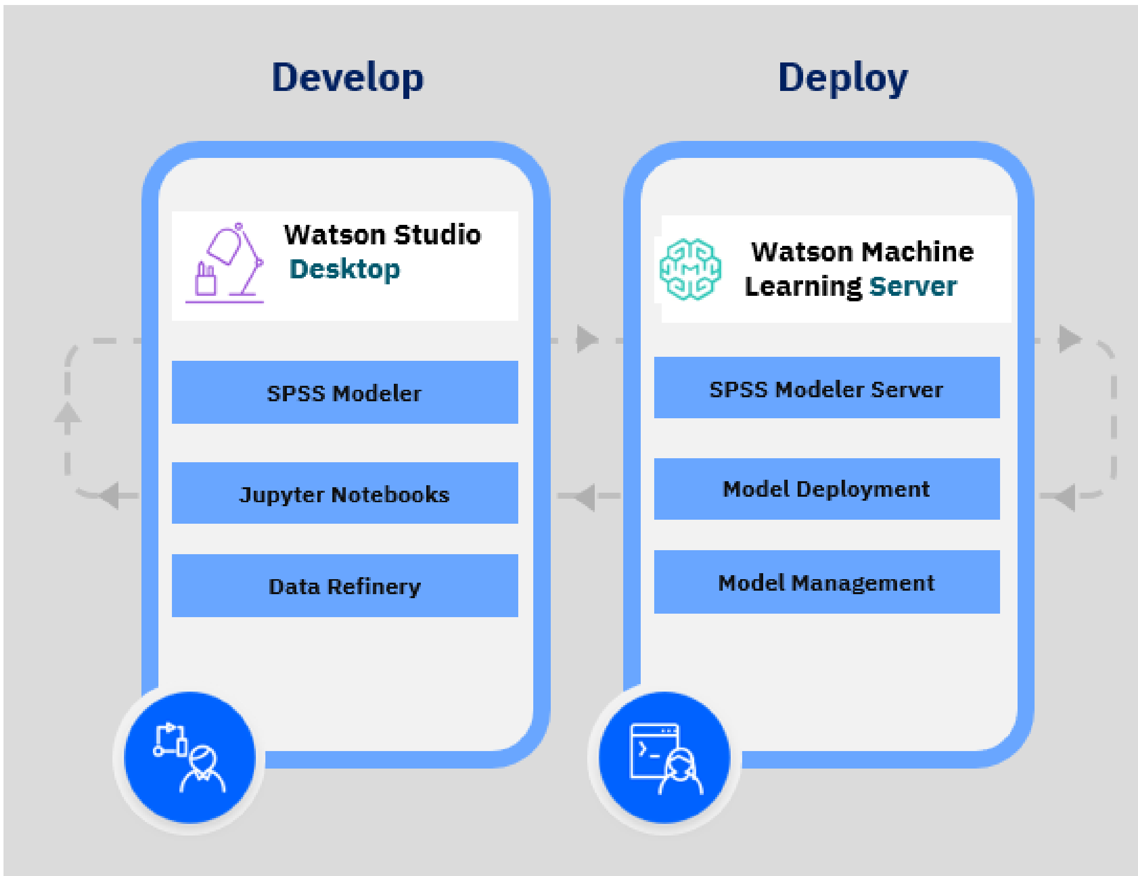
Files for this lab are in the *WSD + WML Server* directory. In the lab we will refer to this folder as the *git-repo* folder.

Introduction

Watson Studio Desktop (WSD) is a comprehensive data science workbench. WSD builds on the success of **SPSS Modeler**, the leading visual data science workbench that has been used by thousands of customers for three decades.

WML Server is a platform for deploying models. It supports deployment of SPSS as well as open source models.

WSD and WML Server support the entire machine learning lifecycle, which we will demonstrate in this lab.



Part 1: Build a Churn Model

In this section we will build a customer churn model in the Flows interface of WSD.

Use Case

Goal: Identify who is likely to churn from a Telecommunications company.

Approach:

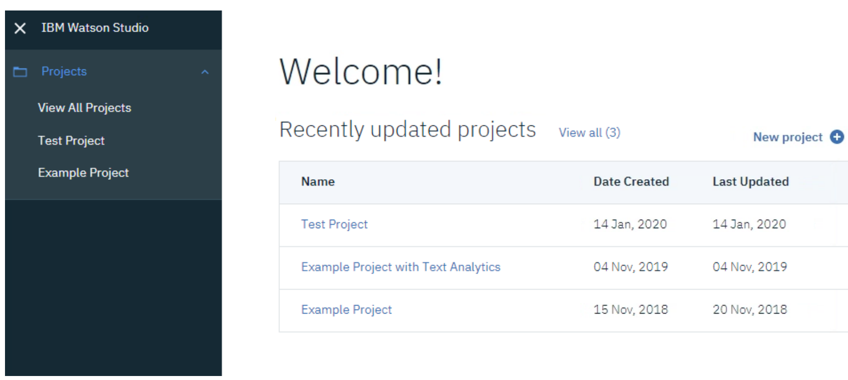
- Use data extracts from three separate systems within the company
- Join the three data sets together
- Derive any fields which may add value
- Choose the appropriate modeling technique
- Automatically generate a model to identify who is likely to churn
- Review results

Benefit: Identify those likely to leave the company so that pre-emptive action can be taken, if so desired.

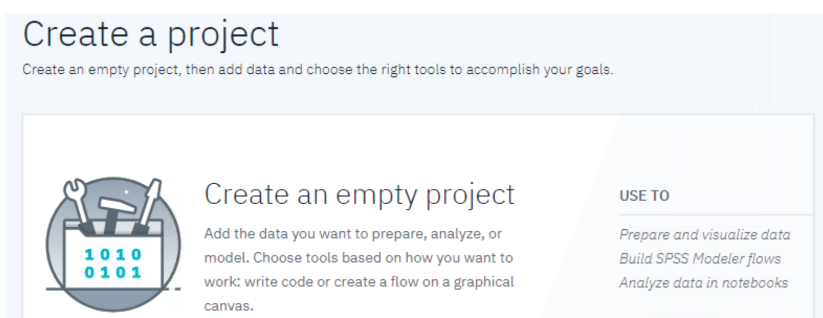
Create a Project and Load the Data

1. Open **WSD**.

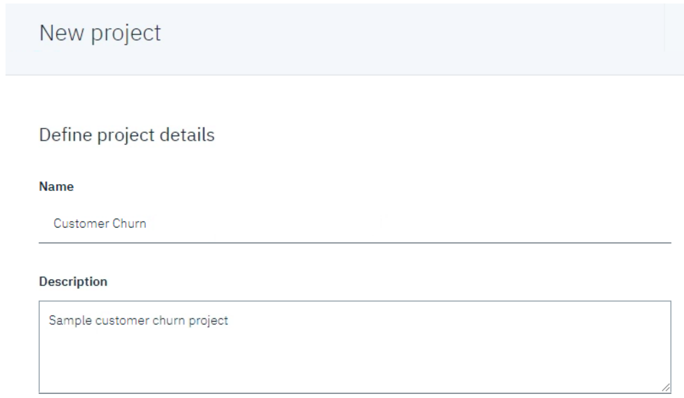
2. From the **Projects** home page, click **New Project.**



3. Click **Create an empty project**.



4. Enter a **Project Name** and, if you wish, a **Project Description**.



5. Click **Create**.

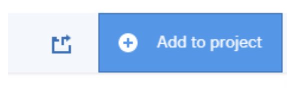
Next, we need to upload model training data. The assets can be found within the *Churn Prediction* subfolder of your *git repo* folder.

There are two options when using csv files within WSD. You can choose to import the data file as:

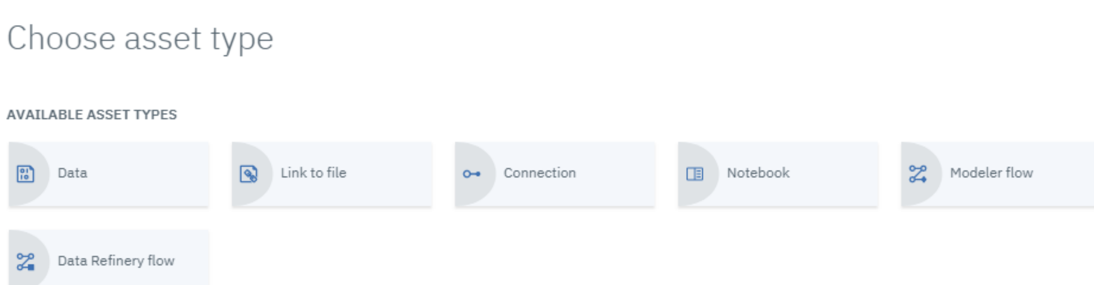
- **Data**: This option imports the file into WSD and stores the file along with all other project assets on the file system.
- **Link to File:** This option connects to a file on your existing local file system and reads the file from that location each time the flow is executed.

In this lab we will import the files as **Data**.

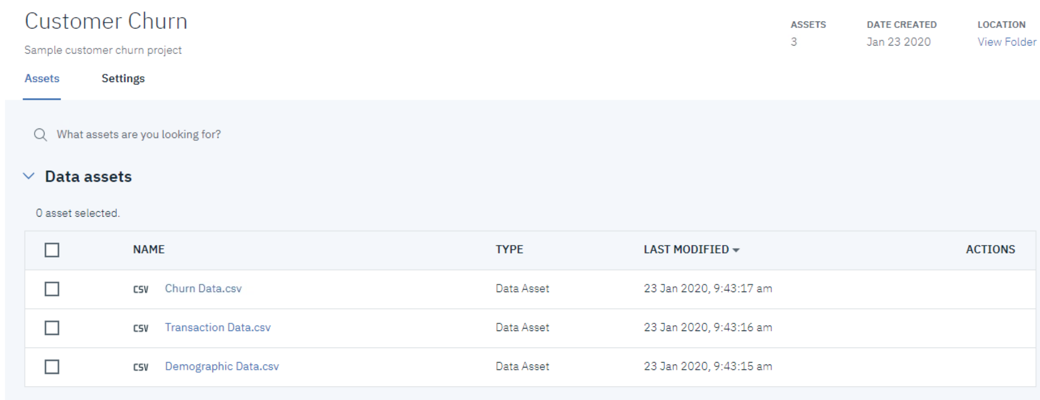
6. From the **Project Asset** page, choose **Add to Project**.



7. Select **Data**.

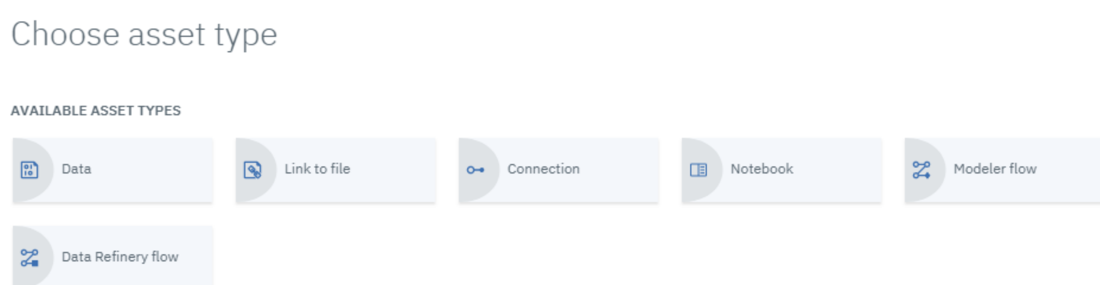


- Click the **Browse** link
- Navigate to the downloaded *Churn Prediction* folder and select the three files
    - *Churn Data.csv*
    - *Transaction Data.csv*
    - *Demographic Data.csv*
- Click **Open.**

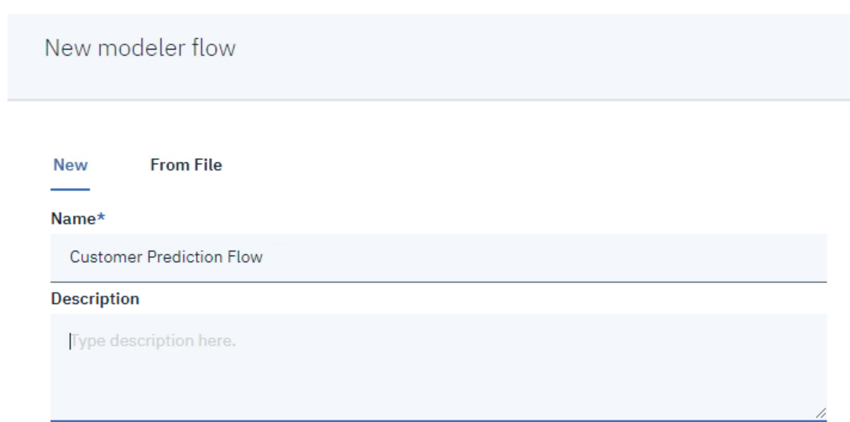


Create the Churn Model: Step-By-Step Guide

8. Click **Add to Project,** select **Modeler Flow**.

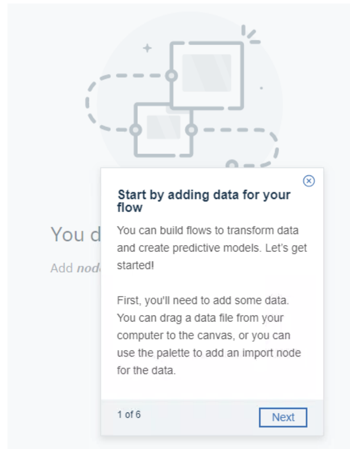


9. Enter flow name.



10. Click **Create.**

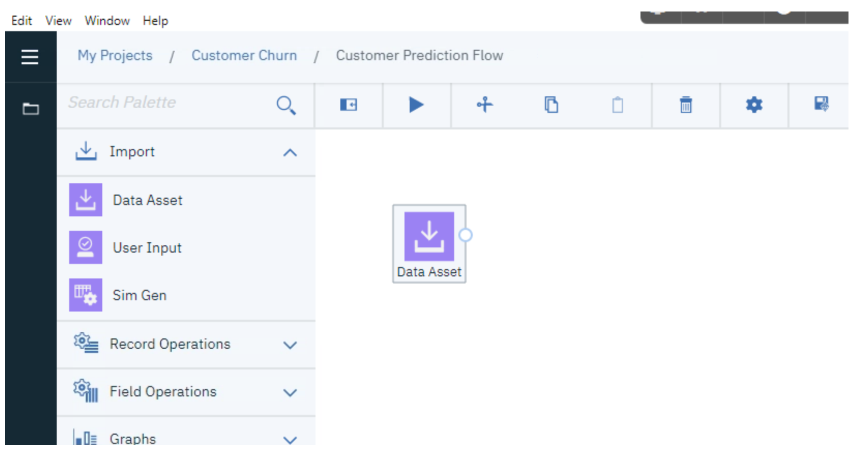
11. Close the **Welcome** wizard.



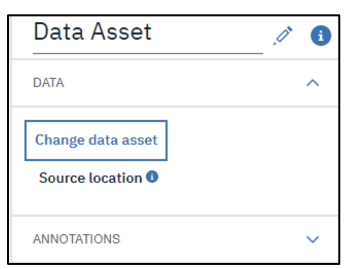
Connect to the Required Data Files

The flow is opened with a blank canvas, therefore the first thing required is to add some data to the flow.

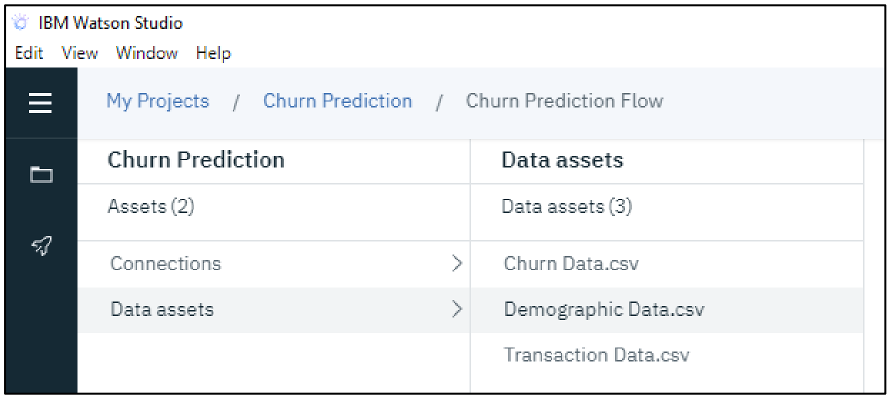
12. Expand the **Import** tab and drag the **Data Asset** node to the canvas.



13. Double-click on the **Data Asset** node so the **Data Asset** settings are displayed.



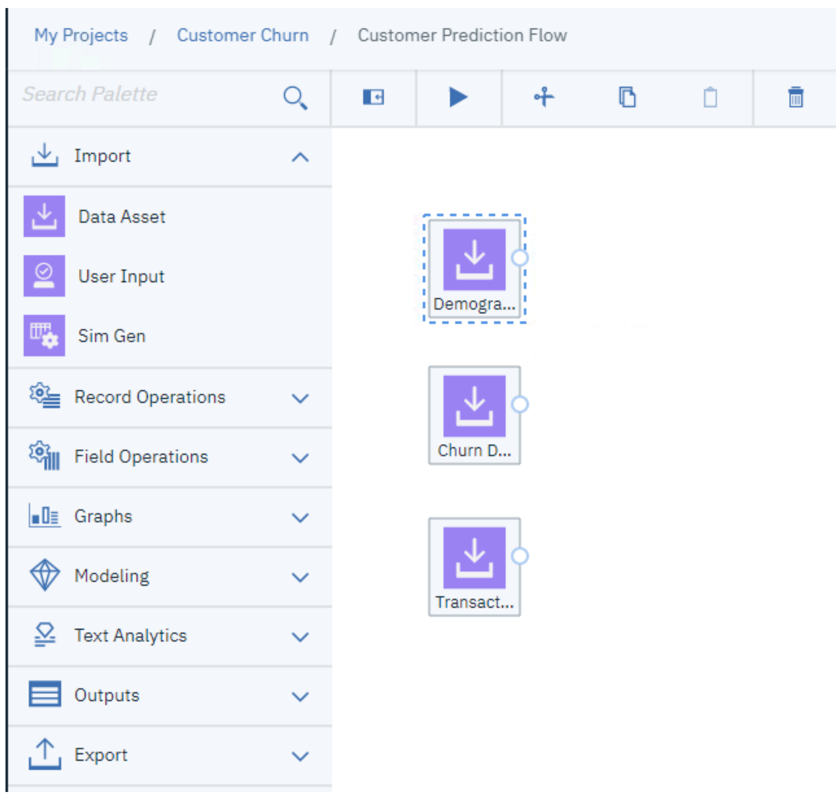
14. Click **Change Data Asset**. Select *Data Assets - > Demographic Data.csv.*



Click **OK**, then click **Save.**

15. Repeat steps **5-7** to add data nodes for
    a. *Transaction Data.csv*
    b. *Churn Data.csv*

The flow should now look similar to the following screenshot.



Next, preview data for each data source.

16. Right click on one of the data asset nodes (e.g. *Demographic* **Data Asset**) node and choose **Preview** from the menu.

ID	Sex	Region	Children	Est_Income	AgeGroup	Car_Owner	Status
1.000	F	2.000	1.000	38000.000	1.000	N	S
6.000	M	3.000	2.000	29616.000	3.000	N	M
8.000	M	1.000	0.000	19732.800	3.000	N	M
11.000	M	3.000	2.000	96.330	4.000	N	S

17. Repeat the data **Preview** for the other data sets.

- *Transaction data* contains 8 fields of data: transactional information such as payment method, number of visits to the store, monthly spend etc.
- *Churn Data* contains only two fields – the ID field (which is available in all three files) and an indicator showing whether that customer has left the company.

Note: *If the table is wider than the preview window, you will need to scroll to the bottom of the record-set (10 records) and then access the scroll bar which allows you to scroll left & right.*

Next, we'll merge the data files.

Data sets can be joined in two ways:

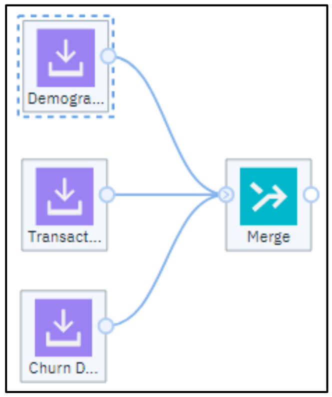
Merge: Add fields (columns) to an existing data set either by matching on a key or by order of records

Append: Add records (rows) to an existing data set – usually the two files contain the same columns (and column names) but this does not have to be the case

Our three data sets all have a common field (*ID*) which can be used to join the three fields together using a **Merge** node.

18. From the **Record Operations** palette, select a **Merge** node and drag it on to the canvas.

19. Connect the three source nodes to the **Merge** node.



20. Double click on the **Merge** node to access the settings.

- Expand the **Merge Method** tab and change the method from *Order* to *Keys*.
- Click the **Add Columns** button to access the field selection options.
- Select the field *ID* and click **OK**.

21. Click **Save** to close the settings option and return to the flow.

Explore the Merged Data

Now we can work with the merged data. Modeler provides several ways to explore the data.

22. Right-click the merge node and select **Profile**.

    Notice that the **Profile** output has three sections:

    Data: This screen is similar to the preview discussed previously, but shows the first 200 records of the data set.

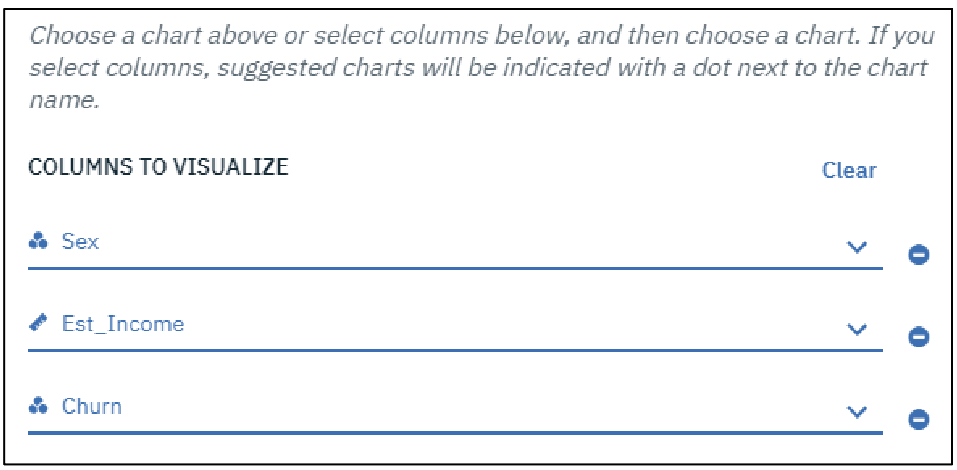
    Profile: Shows the basic distribution of each field, along with the data type and some summary statistics about the distribution.

    Note: the summaries statistics are based ONLY upon the 200 record sample, and not the full data set.

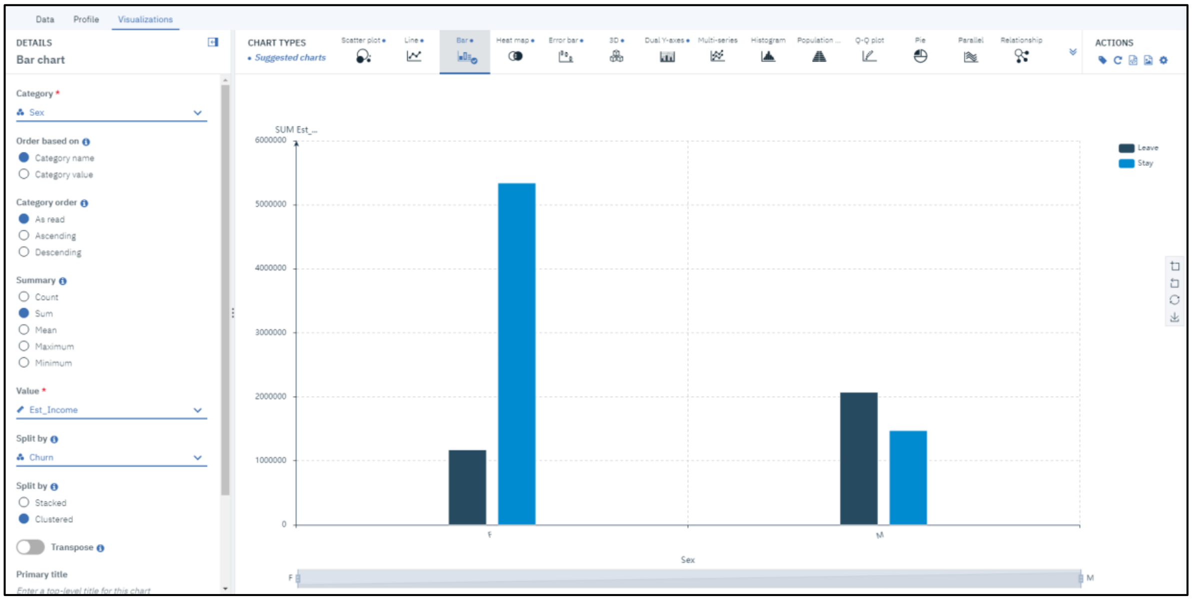
    Visualizations: This allows the user to create graphs/charts of the various fields in the dataset, or combinations of fields

23. Click the **Visualization** tab.

24. Add the fields *Sex*, *Est_Income* and *Churn* to the list of chart fields and click the **Visualize Data** button at the bottom of the dialog.



25. The default chart displayed a clustered scatterplot, which is not ideal for this combination of fields. From the top of the dialog box, change the chart type to be a *Bar Chart.*



This graph shows that there is most likely a relationship between *Income* & *Gender* with *Churn*

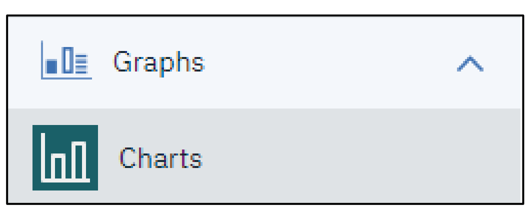
- For males: customers who leave have higher income than customers who stay
- For females: customers who stay have higher income than customers who leave.

26. If you wish, reset the visualization pane and explore other fields.

27. When the exploration is completed, return to the canvas using navigation in the menu bar.

Visualizations in the **Profile** view are useful for exploration. If we want to persist (or view) visualizations every time the flow runs, we will need to use the **Graph** nodes.

28. From the **Graphs** palette, select the **Charts** node and drag it to the canvas.



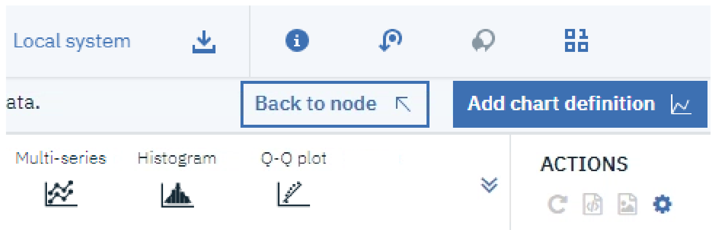
29. Connect the **Charts** node to the **Merge** node.

30. Double-click the **Charts** node to access the **Settings** pane.

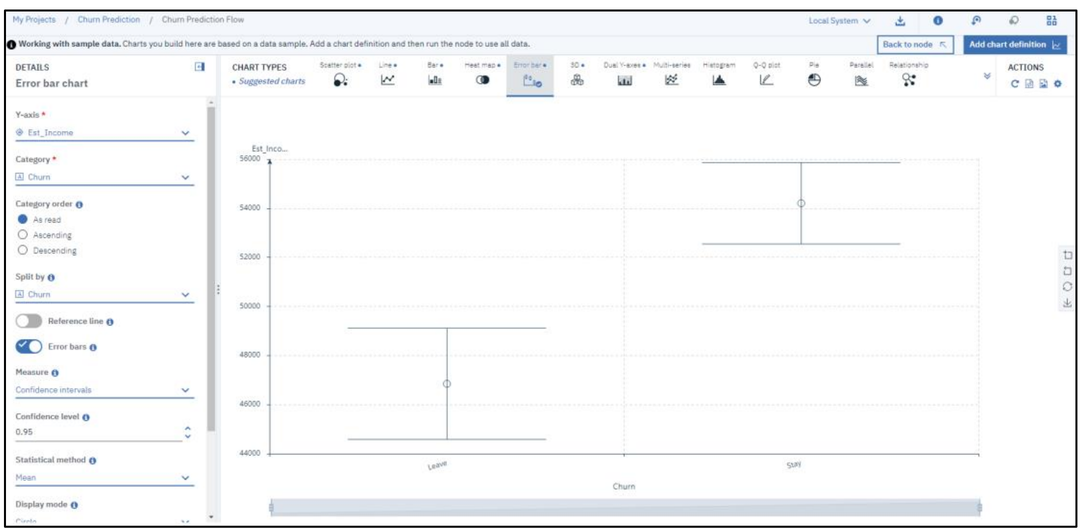
31. Click **Launch Chart Builder.**

    Notice that this is the same chart builder as in the **Profile** option. Visualizations can be built in the same way, the main difference is the ability to persist the chart definition.

32. Recreate the same bar chart and click **Add Chart Definition.**



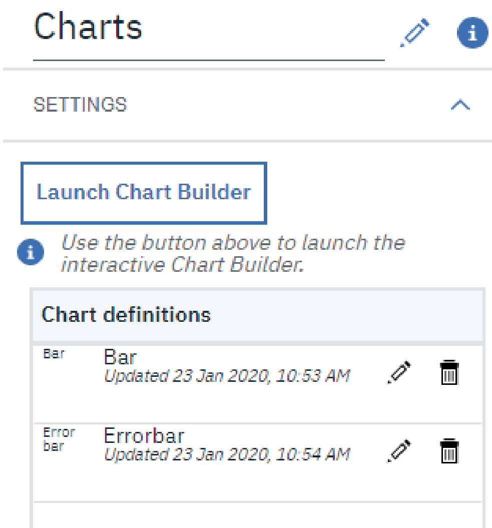
33. Click the **Start Over** option ( ) at the top-right, and create a second chart. For example, an *Error Bar Chart* of *Average_Spend_Per_Store_Visit* and *Churn*.



34. Again, click **Add chart definition.**

35. When both definitions have been created, click **Back to Node** in order to close the chart builder.

The chart settings now have two definitions. Click **Save**.

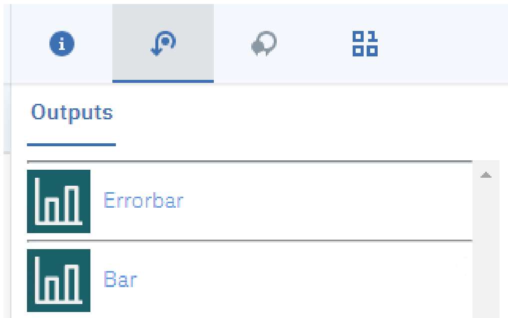


Let's review the output of the **Chart** node when we run the flow.

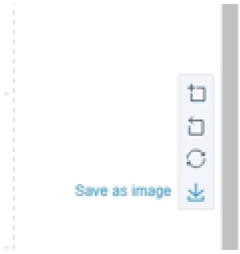
36. Right click on the **Chart** node in the flow and select **Run**.

37. Navigate to the **Outputs** view.

Double click on each graph to display it.



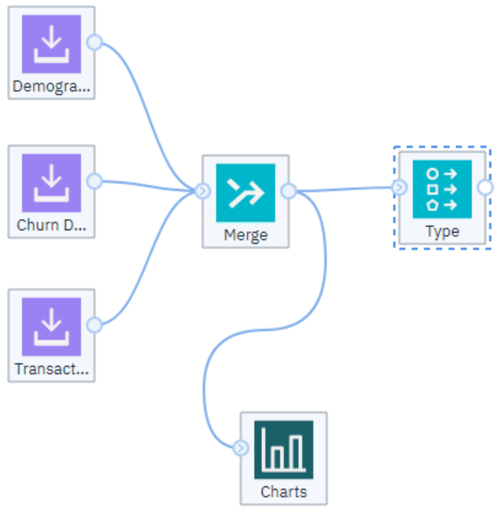
Notice that you can save each graph as an image.



Return to canvas.

38. From the **Field Operations** palette, select a **Type** node.

39. Connect the **Merge** node to the **Type** node.



40. Double-click on the **Type** node to edit it.

The **Type** node is used to determine how data will be used in subsequent nodes. The **Type** node

- Displays the range of values in each field (*Min/max or category values*)
- Determines the type of data (*Measurement*),
- Can be used to specify input or target fields for a model (*Role*)
- Provides several data validation options.

When we first add the **Type** node, information for each field within is not filled out until we invoke the **Read** function.

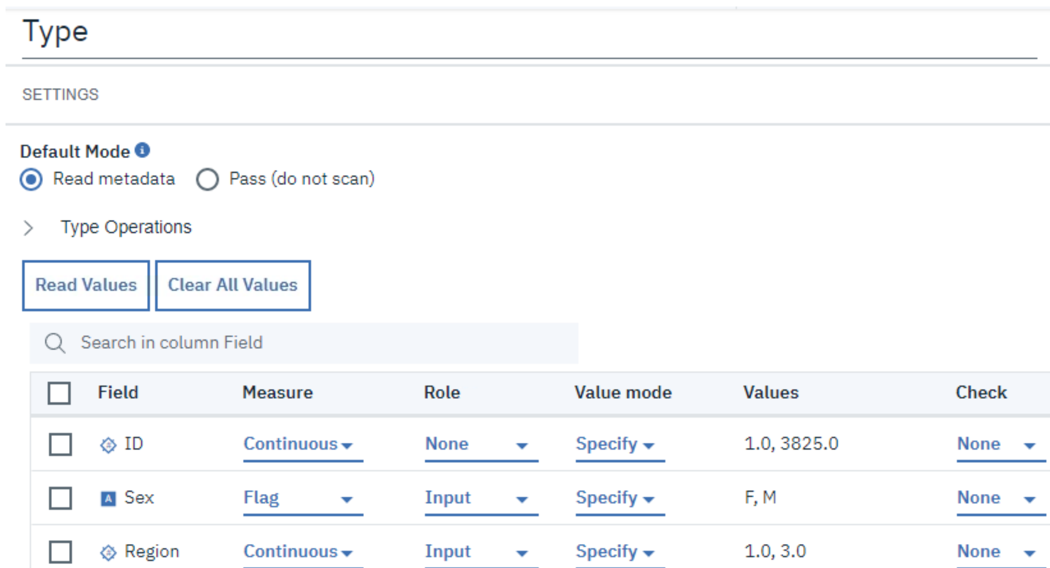
Field	Measure	Role	Value mode	Values	Check
ID	Continuous	Input	Read		None
Sex	Categorical	Input	Read		None
Region	Continuous	Input	Read		None
Children	Continuous	Input	Read		None
Est_Income	Continuous	Input	Read		None
AgeGroup	Continuous	Input	Read		None
Car_Owner	Categorical	Input	Read		None
Status	Categorical	Input	Read		None

41. Click the **Read Values** button at the top of the **Type** node settings.

In most cases Modeler correctly determines data types. It’s also able to determine the Role of a field (input or target).

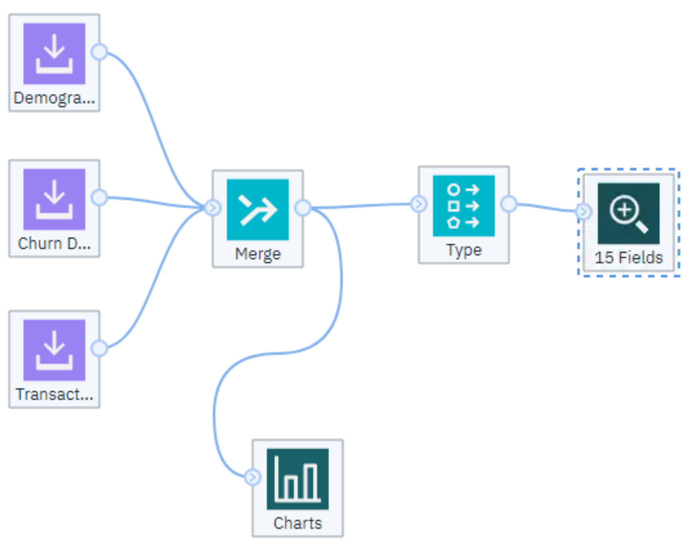
In some cases you may need to change the settings in the **Type** node.

Change the *Role* of the *ID* field to *None* and the *Role* of the *Churn* field to *Target.*



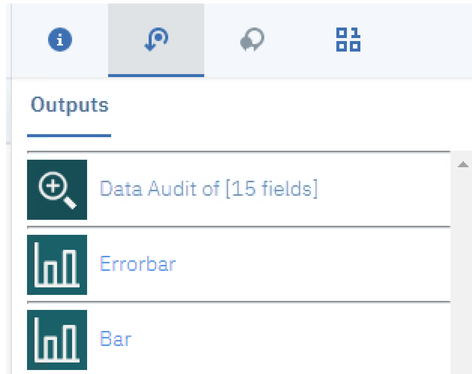
42. Click **Save** to save the **Type** node information.

43. From the **Outputs** palette select the **Data Audit** node and connect it to the **Type** node.

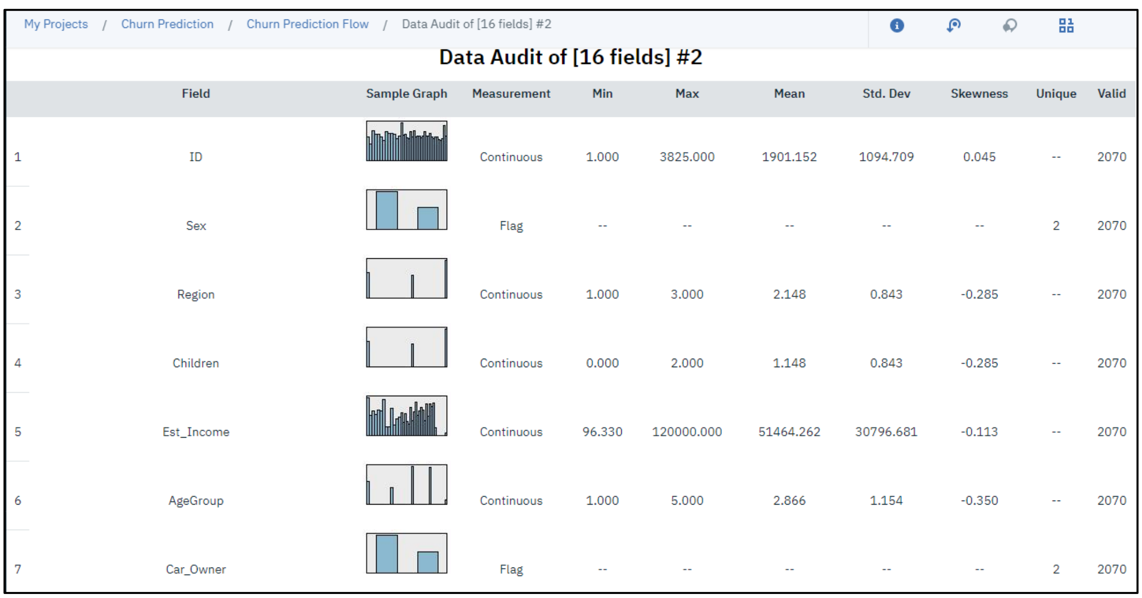


44. Right click on the **Data Audit** node and click **Run**.

45. Double click on the *Data Audit* output from the **Outputs** view.



The data audit output shows an appropriate distributional chart for each field, along with information about the *Measurement Level* and summary statistics for the distribution (based upon the full data set). The final column shows the number of Valid records in the file, which is the total number of non-missing records.



My Projects / Churn Prediction / Churn Prediction Flow / Data Audit of [16 fields] #2

Data Audit of [16 fields] #2

	Field	Sample Graph	Measurement	Min	Max	Mean	Std. Dev	Skewness	Unique	Valid
1	ID		Continuous	1.000	3825.000	1901.152	1094.709	0.045	--	2070
2	Sex		Flag	--	--	--	--	--	2	2070
3	Region		Continuous	1.000	3.000	2.148	0.843	-0.285	--	2070
4	Children		Continuous	0.000	2.000	1.148	0.843	-0.285	--	2070
5	Est_Income		Continuous	96.330	120000.000	51464.262	30796.681	-0.113	--	2070
6	AgeGroup		Continuous	1.000	5.000	2.866	1.154	-0.350	--	2070
7	Car_Owner		Flag	--	--	--	--	--	2	2070

46. Scroll down to the bottom part of the **Data Audit** output to find a more detailed listing of the missing values analysis.

    By default, valid values only identify System Missing values (or NaN values) for numeric fields. If you wish to identify other types of missing numeric values, or if you are working with string fields where blank values or whitespaces are considered missing values, you will need to configure it in the **Type** node.

    From the output we can see that all fields have 2070 valid values, which is the total number of records in the file, therefore there are no missing values in our datasets.

	Field	Measurement	Outliers	Extremes	Action	Impute Missing	Method	% Complete	Valid Records	Null Value
1	Sex	Flag	--	--	--	Never	Fixed	100.000	2070	0
2	Region	Continuous	0	0	None	Never	Fixed	100.000	2070	0
3	Children	Continuous	0	0	None	Never	Fixed	100.000	2070	0
4	Est_Income	Continuous	0	0	None	Never	Fixed	100.000	2070	0
5	AgeGroup	Continuous	0	0	None	Never	Fixed	100.000	2070	0
6	Car_Owner	Flag	--	--	--	Never	Fixed	100.000	2070	0
7	Status	Nominal	--	--	--	Never	Fixed	100.000	2070	0
8	Churn	Flag	--	--	--	Never	Fixed	100.000	2070	0
9	Paymethod	Nominal	--	--	--	Never	Fixed	100.000	2070	0

*Note: In the current version of WSD, interactive analysis it not available, but when it is then there will be options to automatically **Impute missing values** and replace them with a either a Fixed or Random value, and expression or even calculate them based upon a Decision Tree algorithm.*

47. Return to the canvas.

Data Preparation and Feature Creation

Deriving new features for modeling is a typical step of machine learning lifecycle. In this section we will create two new features, *Total Spend* and *Total Spend Difference From Segment*.

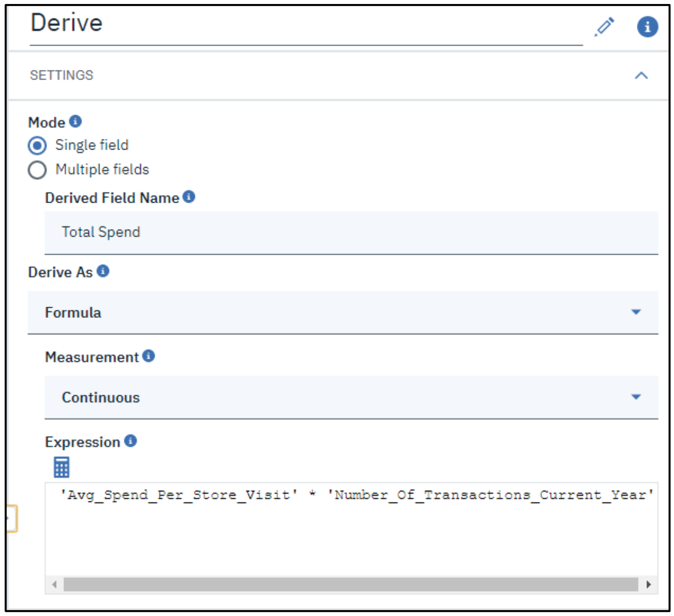
48. From the **Field Operations** palette select a **Derive** and connect it to the **Type** node.

49. Double-click the **Derive** node to edit the settings.

50. Edit the **Derive** node so that it has the following settings:

- **Derived Field Name:** *Total_Spend*
- **Derive As:** *Formula*
- **Measurement:** *Continuous*
- **Expression:** 'Avg_Spend_Per_Store_Visit' * 'Number_Of_Transactions_Current_Year'

The completed dialog should be as follows:



51. Click **Save**.

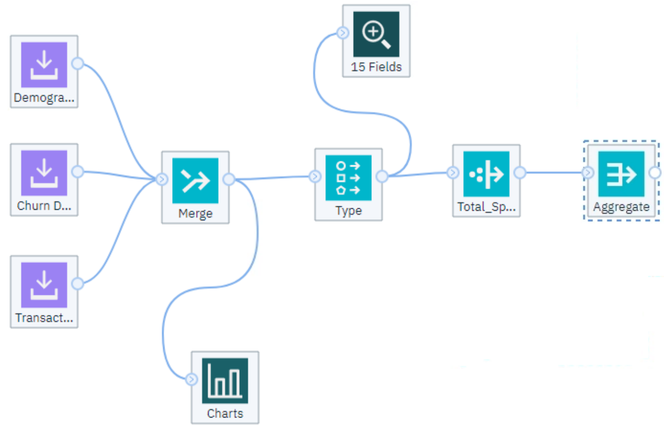
52. If you want to check that the derived field has worked successfully, right click on **Derive** and click **Preview**.

    This will show you the results or an error if you have not defined the expression correctly.

Number_Of_Transactions_Current_Year	Total_Spend
12.000	2472.960
15.000	682.500
12.000	269.280
13.000	427.440

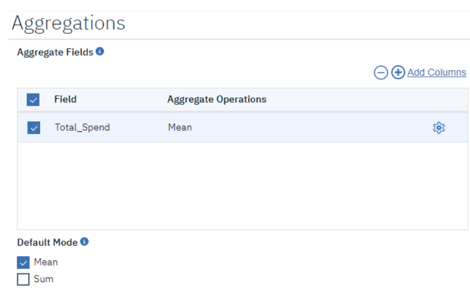
We will now use the *Total_Spend* field to calculate how different each customer is from the average total spend of their customer segment. This will involve aggregating *Total_Spend* by *Customer Segment* (to calculate the mean for each segment) and then deriving a difference from this mean value.

53. From the **Record Operations** palette, select an **Aggregate** node and connect it to the **Derive** node.



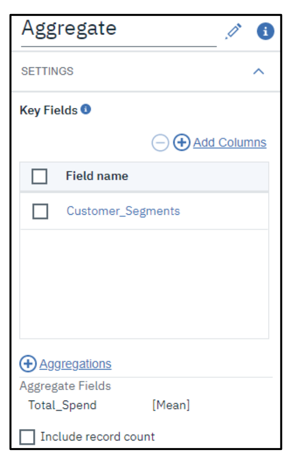
54. Double-click the **Aggregate** node to edit it. Change settings in the **Aggregate** node:

- **Key Fields:** *Customer_Segments*
- **Aggregations:** Add the field *Total_Spend* and the *Mean* value



- **Include Record Count:** Uncheck the box

  The settings should look similar to this screenshot:

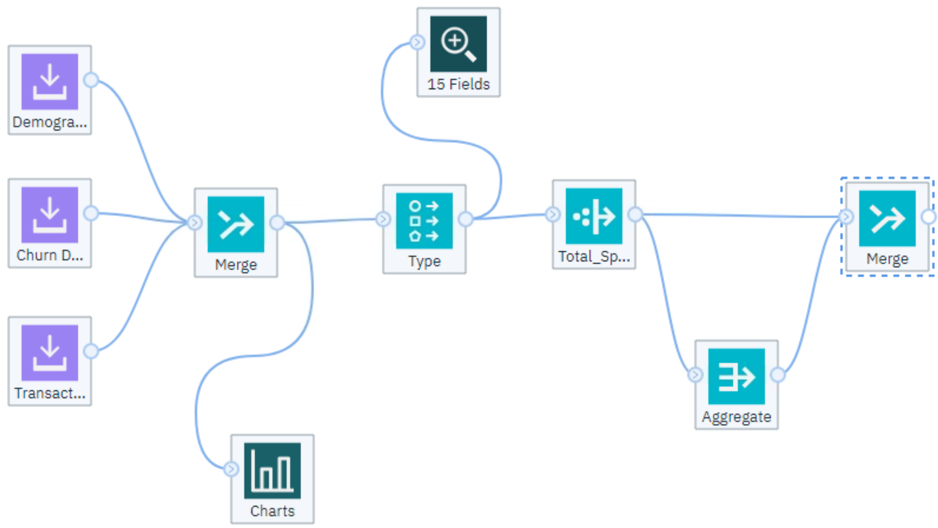


55. Click **Save.**

The aggregated data needs to be joined to the rest of the customer data.

56. From the **Record Operations** palette, select a **Merge** node and connect it to the *Total_Spend* **Derive** node.

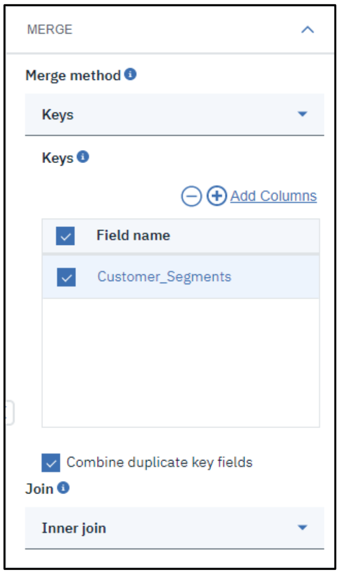
57. Connect the **Aggregate** node to the **Merge** node.



58. Double-click the **Merge** node to edit it.

59. Expand the **Merge** section of the settings and make the following changes:

- **Merge Method:** *Key*
- **Field Name:** *Customer_Segments*
- **Join Type:** *Inner Join*



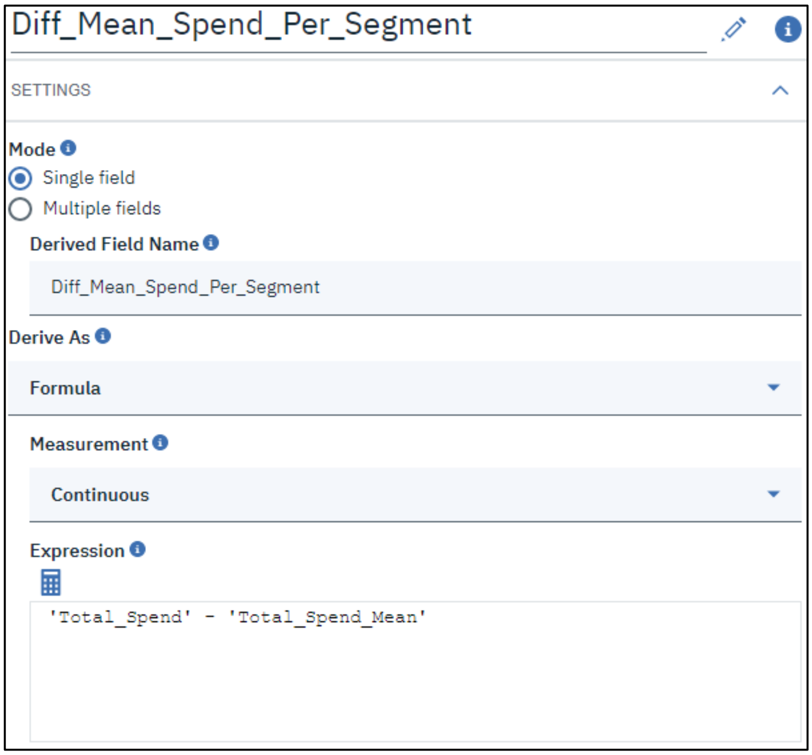
60. Click **Save**.

61. From the **Field Operations** drag another **Derive** node and place it on the canvas to the right of the **Merge** node.

62. Connect from the **Merge** node to the **Derive** node.

63. Edit the **Derive** node so that it has the following settings:

- **Derived Field Name:** *Diff_Mean_Spend_Per_Segment*
- **Derive As:** *Formula*
- **Measurement:** *Continuous*
- **Expression:** *'Total_Spend' - 'Total_Spend_Mean'*



64. Click **Save.**

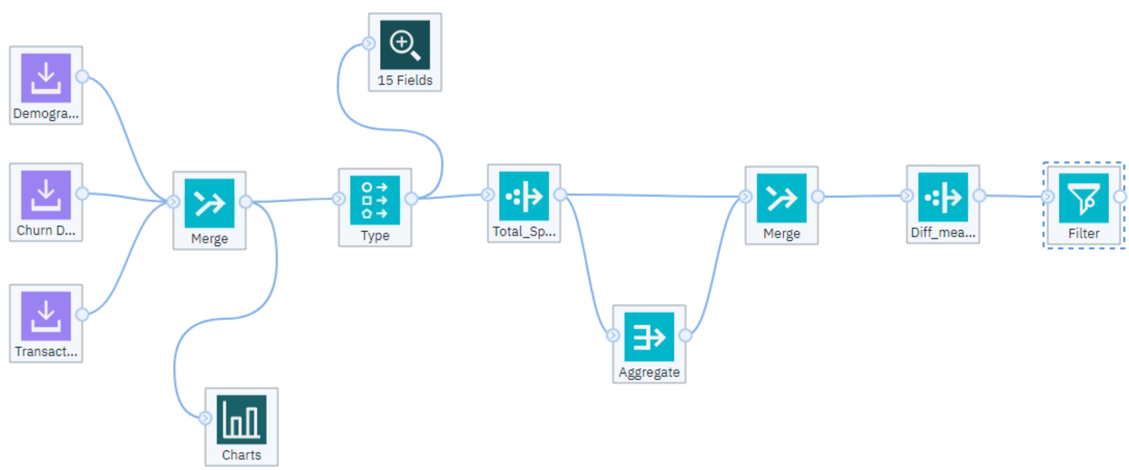
65. Right-click on the last **Derive** node and select **Preview** to see the current state of the dataset.

Scroll all the way to the right. You will see the three new fields created with **Derive** an **Aggregation** nodes.

Total_Spend	Total_Spend_Mean	Diff_mean_spend_per_segment
609.630	570.228	39.402
161.980	570.228	-408.248
1138.720	570.228	568.492
0.000	570.228	-570.228
0.000	570.228	-570.228
328.380	570.228	-241.848

The *Total_Spend_Mean* was an interim field used for calculation purposes. This field is no longer required and therefore can be removed from the dataset.

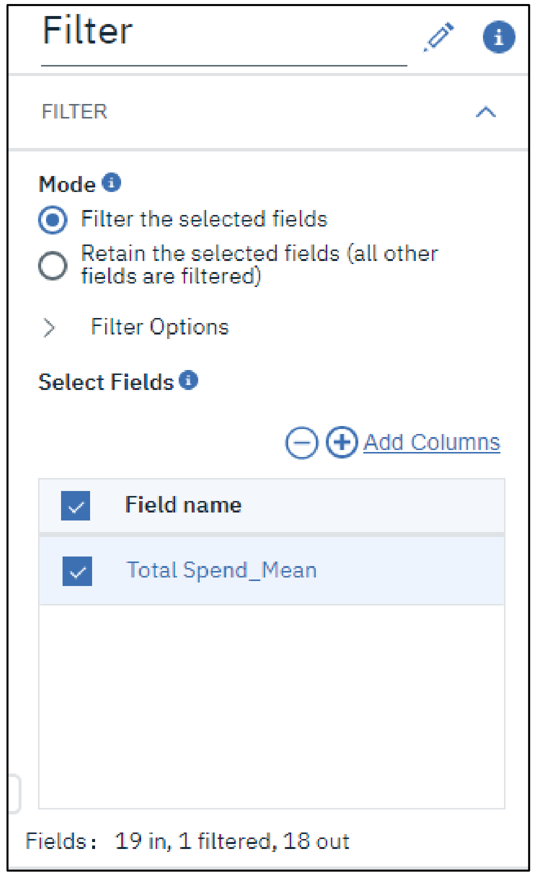
66. From the **Field Operations** palette, select a **Filter** node and connect it the second **Derive** node.



67. Double-click the **Filter** node to edit it.

68. Change the **Filter** node options:

- **Mode:** *Filter the selected fields*
- **Add Columns:** *Total_Spend_Mean*



69. Click **Save**.

Building a Predictive Model

Next, we'll build a model to predict customer churn. Modeler includes several types of classification algorithms. In this lab we will use a popular decision-tree based model, *C5*.

70. From the **Field Operations** tab add the **Type** node and connect it to the **Filter** node.

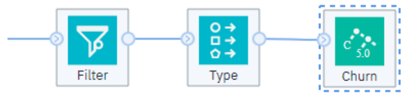
    We need to add the **Type** node to configure the fields that were created by the **Derive** node.

71. Double click on **Type** node. Click the **Read Values** button.

72. Click **Save** to close the **Type** node.

73. From the **Modeling** palette select a C5.0 node and connect it to the **Type** node.

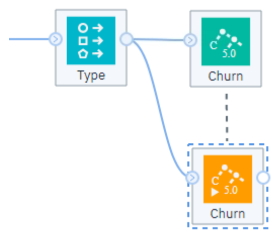
The **C5** node displays *Churn* in the node because we set *Churn* as the *target* field in the **Type** node.



74. Right click on **C5.0** and select **Run**.

This action will build a model – an orange **C5** nugget that appears on the canvas.

The green C5 node is the *algorithm* – it's used to build the model, while the orange **C5** node is the *model*. This model will be used to score new records.



75. Right click the generated model (orange node) and select **Preview**.

76. Scroll all the way to the right of the **Preview** window.

Notice that two new columns created within the data set.

- *$C-Churn*: The model prediction of the target field (prediction of churn)
- *$CC-Churn*: The model confidence where 1 indicates high confidence in the $C-Churn prediction and 0.5 (for a binary target) indicates equal confidence in both categories (no differentiation).

Close the **Preview** window.

Total_Spend	Diff_mean_spend_per_segment	$C-Churn	$CC-Churn
609.630	39.402	Leave	0.800
161.980	-408.248	Leave	0.833
1138.720	568.492	Leave	0.800
0.000	-570.228	Stay	0.925

Next, we'll examine the model.

77. Right-click again on the generated model (orange nugget) node and select **View Model**.

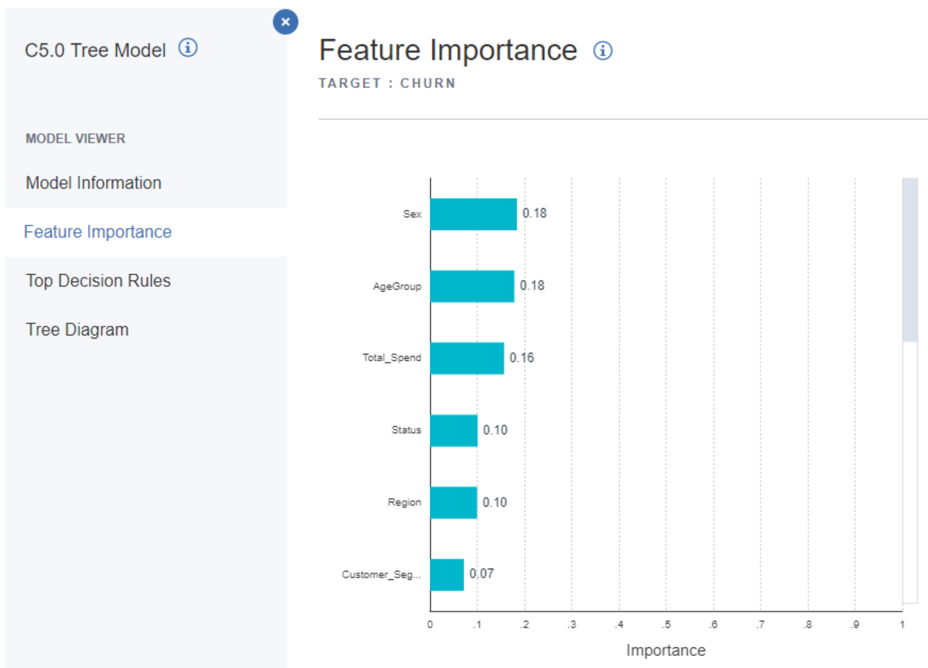
We can review various information about the **C5** mode:

Model Information: Model type, the number of input features, nodes and the maximum levels of the tree (defined in the C5.0 algorithm node)

Feature Importance: Which of the input features are most important in the model (in determining a customer's propensity to churn)

Top Decision Rules: The rule logic which leads to a particular outcome (Mode Category), along with the number and percent of total records in that node and the percentage of customers in that node who have the given outcome.

Tree Diagram: a graphical representation of the *Top Decision Rules*. All the above information can be obtained by hovering the cursor over any of the nodes.

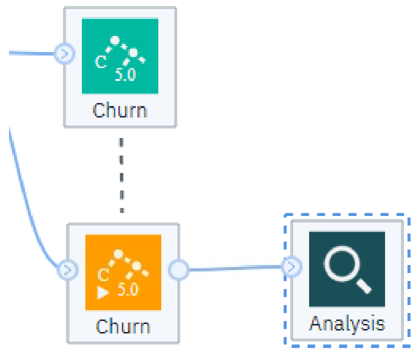


When you have finished exploring the model output return to the canvas.

Evaluate the Model Performance

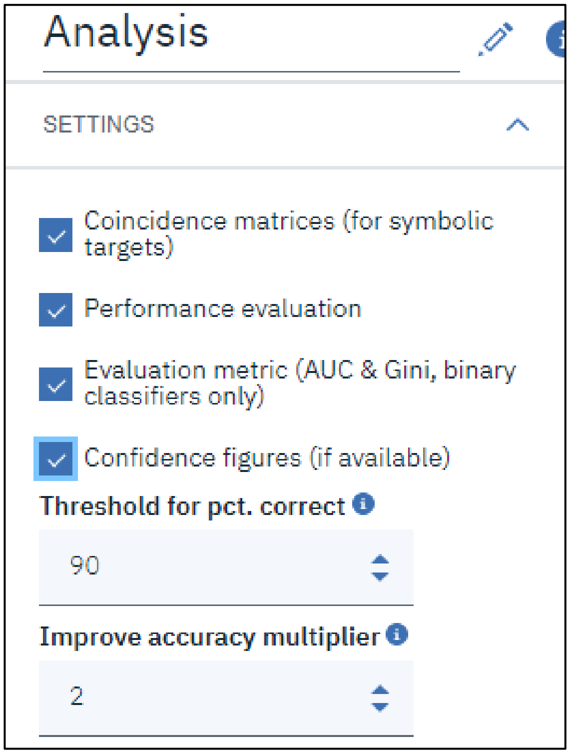
The last step in model development is model evaluation. There are two nodes generally used for model evaluation in Modeler Flows: **Analysis** and **Evaluation**.

78. From the **Output** palette select the **Analysis** node and connect it to the model nugget.



79. Double-click the **Analysis** node to edit the **Settings**.

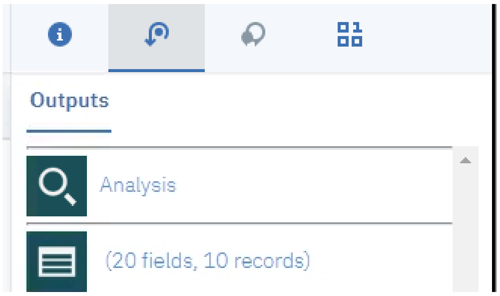
80. Check the four checkboxes at the top of the **Settings** tab.



81. Click **Save** to return to the canvas.

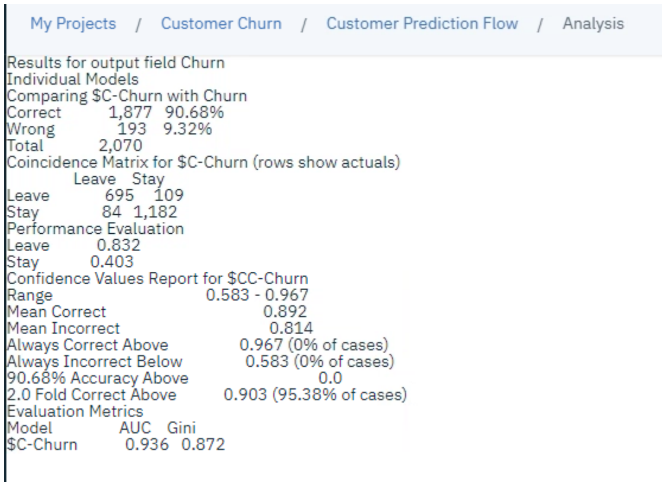
82. Right click the **Analysis** node and select **Run** from the context menu.

83. In the **Output** view double-click the **Analysis** output.



The **Analysis** node shows the following:

- The number of correct and incorrect predictions
- Overall model accuracy values (85.36%)
- Coincidence Matrix (also called confusion or classification matrix)
- Other statistics about the level of confidence to which the model always classifies correctly as well as the Gini and AUC coefficients.



My Projects / Customer Churn / Customer Prediction Flow / Analysis

Results for output field Churn
Individual Models
Comparing $C-Churn with Churn

Correct	1,877	90.68%
Wrong	193	9.32%
Total	2,070	

Coincidence Matrix for $C-Churn (rows show actuals)

	Leave	Stay
Leave	695	109
Stay	84	1,182

Performance Evaluation

Leave	0.832
Stay	0.403

Confidence Values Report for $CC-Churn

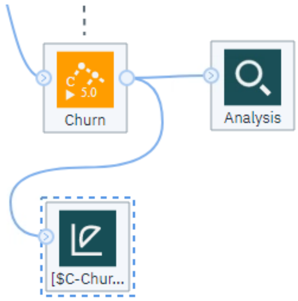
Range	0.583 - 0.967
Mean Correct	0.892
Mean Incorrect	0.814
Always Correct Above	0.967 (0% of cases)
Always Incorrect Below	0.583 (0% of cases)
90.68% Accuracy Above	0.0
2.0 Fold Correct Above	0.903 (95.38% of cases)

Evaluation Metrics

Model	AUC	Gini
$C-Churn	0.936	0.872

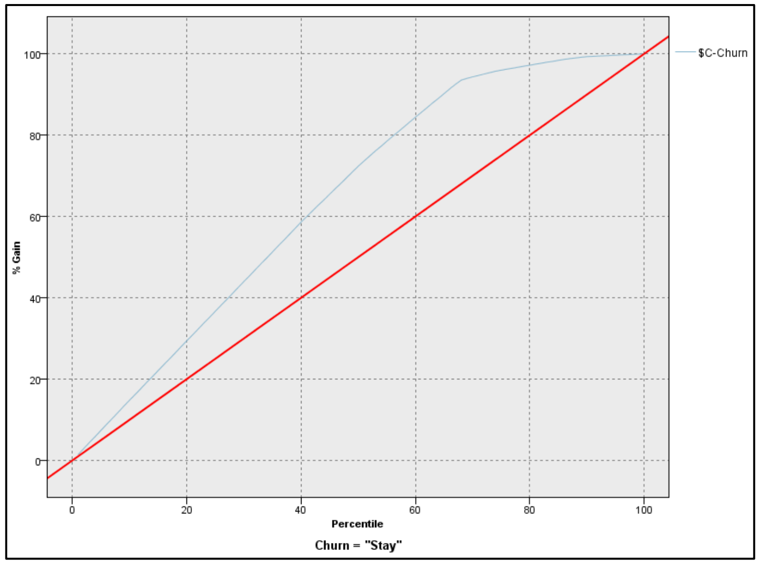
Return to the canvas.

84. From the **Graphs** palette add the **Evaluation** node and connect it to the model nugget.



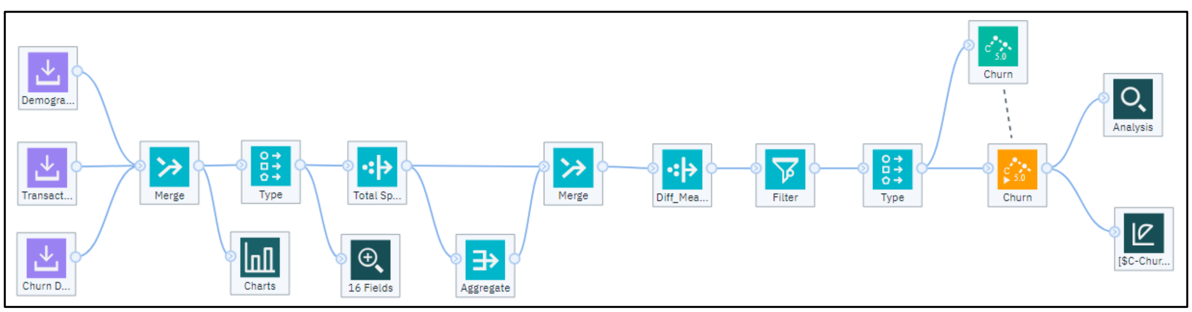
85. Right click on the **Evaluation** node and select **Run**.

86. From the **Output** view open the **Evaluation** output.



The *Gains Chart* shows the uplift, over and above a random chance (red line) that the model provides as you go deeper into the customer list after it has been rank-ordered from highest to lowest propensity to churn.

You have finished building the flow. Your flow should now look something similar to the following:



We included the completed flow (*Churn Prediction Flow - Completed.str*) in the *Churn Prediction* subfolder of the git-repo.

Part 2: Deploy flows in WML Server

We need to complete several setup steps before we can deploy flows in WML Server.

- **Connect to the WML server**
- **Create a Deployment Space**
- **Associate a WSD project with a Deployment Space**

Server connection configuration

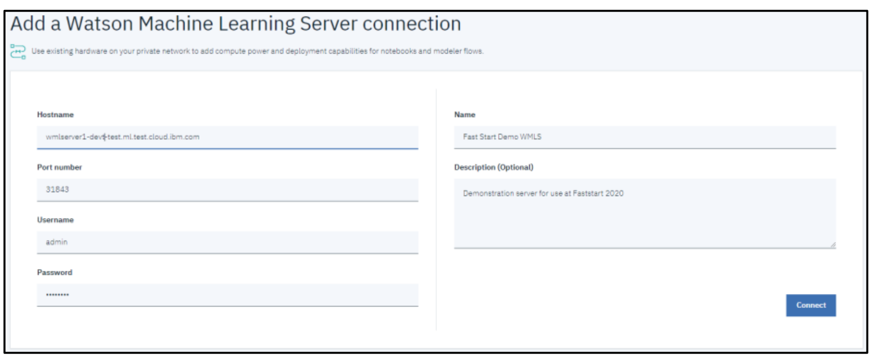
1. Expand the left side bar of WSD by clicking the navigation menu (☰) at the top-left of WSD
   .
2. Click the **Add-ons and services** button.

3. Click **Add machine learning service**

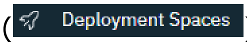
4. Click on the **Machine Learning Server** button

   The lab instructor will provide the following information for the connection:
   - **Hostname**
   - **Port Number**
   - **Username**
   - **Password**

   Enter the connection name.



5. Click **Connect**.

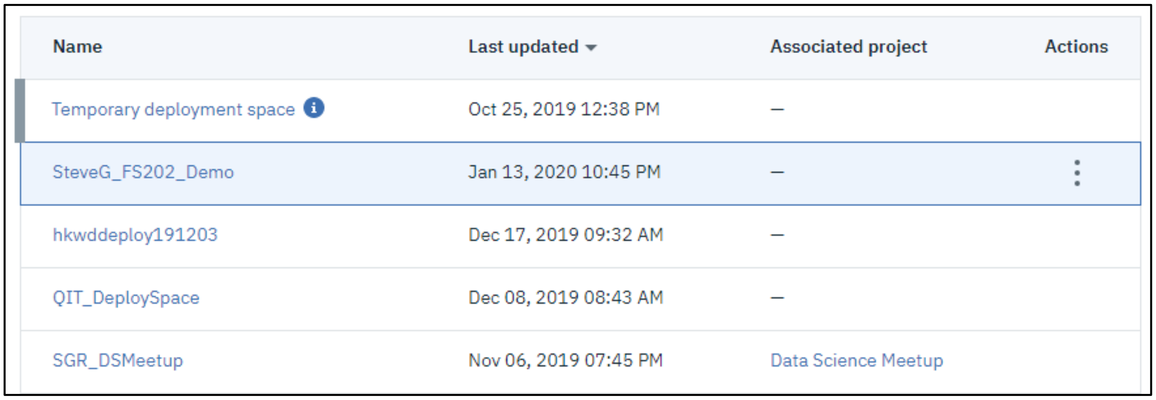
6. From the left pane, click the **Deployment Spaces** link (

)

7. Click **New deployment space**.

8. Provide a unique name for the deployment space (e.g. *SteveG_FS202_Demo*) and click **Create**.

The deployment space is now shown in the list of existing deployment spaces and is ready to be associated with a WSD project



9. From the left pane, click the Project list (

) and select your Churn Prediction project.

10. From the top right of the **Project** summary page, click **Associate a new or existing deployment space.**

11. Select your newly created deployment space and click **Associate.**

The project summary page now shows that the WSD project is associated with the deployment space.

Part 3: Deploy the Churn Prediction Modeler Flow

In this section we will specify a *scoring branch* in Modeler flow, save it in the deployment space, and deploy it for online scoring.

A *scoring branch* is a part of the flow that should be used for scoring data. Some nodes in the flow (such as Data Audit, graphs, evaluation, etc.) are used during model building, but they are not needed for scoring.

1. In WSD open the Modeler flow that you created in **Part 1** of the lab.

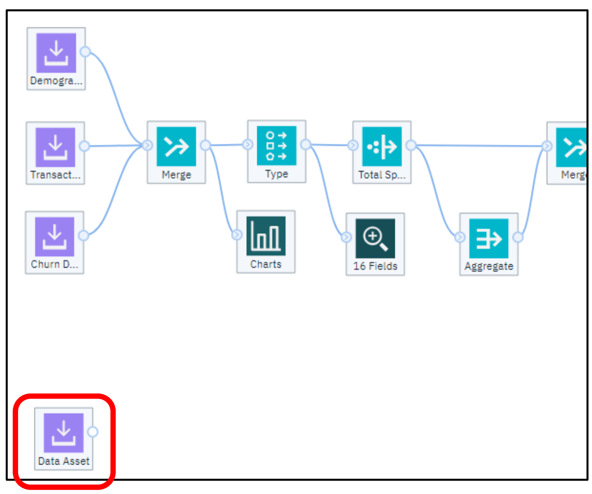
   Note: If you did not Part 1, you can use the Churn Prediction Flow – Completed.str from the Churn Prediction subfolder of the git repo.

   We built the model using training data. Now we need to upload data that we will use for scoring.

2. To upload the dataset from within the Modeler Flow environment, expand the data pane by clicking the button at the top right of the flows environment

3. Click on the **Browse** link, select the file *Churn Batch Input Data.csv* from the git repo folder. Click **Open**.

4. From the **Import** tab select a **Data Asset** node and place it on the canvas.



5. Double click the **Data Asset** node.

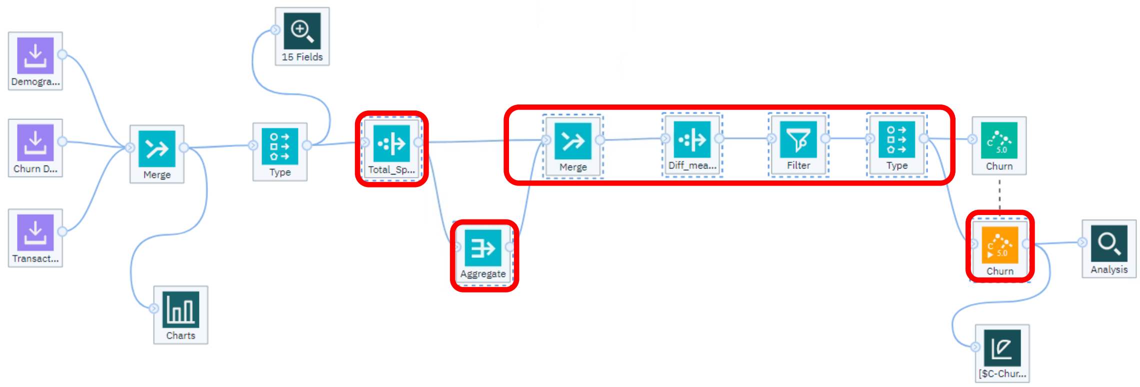
6. Click **Change Data Asset** and select the file *Churn Batch Input Data.csv* from the project **Data Assets**.

7. Preview the data.

Next, we will copy nodes that are needed for scoring.

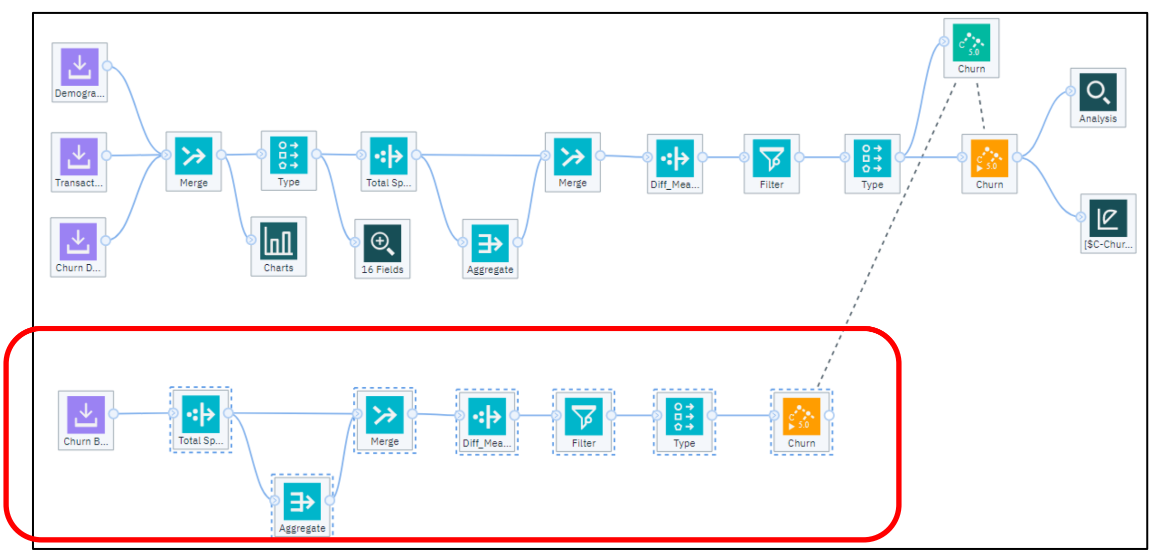
8. Single click the *Total_Spend* **Derive** node to select it.

9. While holding the SHIFT key on the keyboard select all highlighted nodes (from *Total_Spend* to orange model nugget).



10. Select **Copy** from the canvas menu bar, then **Paste**.

11. Connect the Data Asset (scoring data) to the pasted nodes.



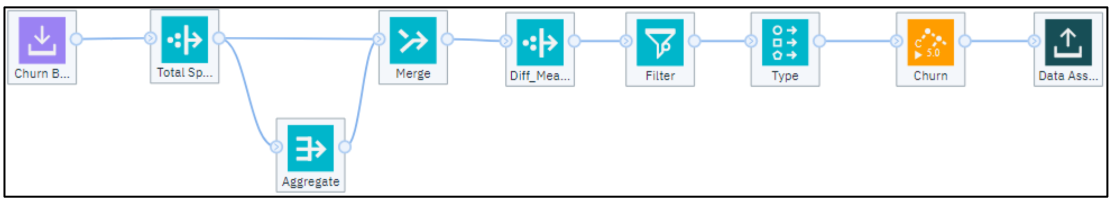
The branch that we just pasted is called the scoring branch – it contains only the nodes that we want to use for scoring. Let's test this branch.

12. Right click on the model nugget (orange node) of the scoring branch and select **Preview**.

When you scroll all the way to the right, you should see the two fields generated by the model *($C-Churn* and *$CC-Churn*)

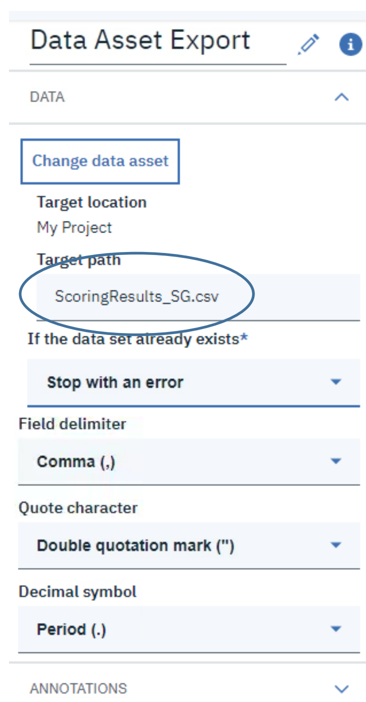
Diff_mean_spend_per_segment	$C-Churn	$CC-Churn
-432.855	Leave	0.833
-557.275	Stay	0.925
-594.835	Stay	0.750

13. From the **Export** palette select the **Data Asset Export** node and connect it to the model nugget of the scoring branch.



14. Double click on the **Export** node.

    In the **Setting** view provide the a unique output file name, for example, *ScoringResults_SG.csv* and click **Save**.



The scoring branch is now configured to read an input file, derive the new features, score the data, and write the results back to the project.

15. Return to the canvas.
16. Right click on the **Data Asset Export** node and select **Run**.

17. Navigate to the project view to verify that scoring results were written to the project.

Deploy Modeler flow for online scoring

In this section we will deploy the flow for real time scoring.

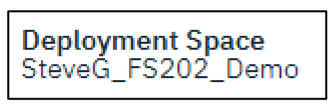
18. If you were in the project view, open the *Customer Churn* flow.

19. Write click on the **Data Asset Export** node and select **Save branch as model.**

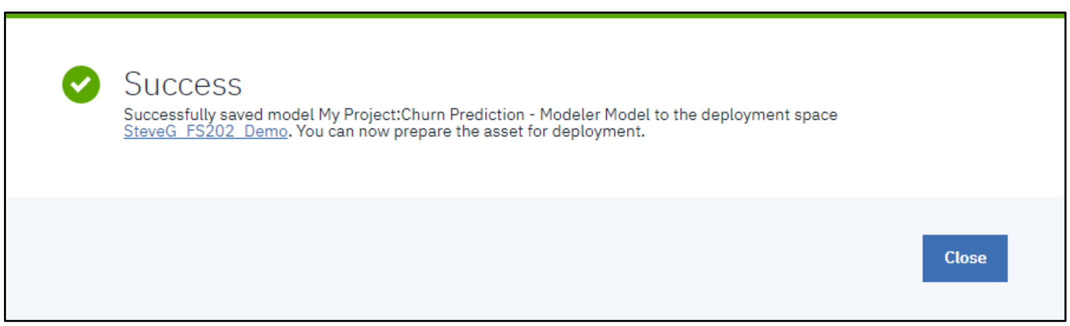
    - Notice at the top of the **Save Model** page, the default selection is to save the Scoring Branch. The **Terminal** node identifying the end of the branch is labeled correctly. We don't need to change it.

    - Change the model name to *Churn Prediction*.

    - The deployment space is already specified because we configured it in the project.

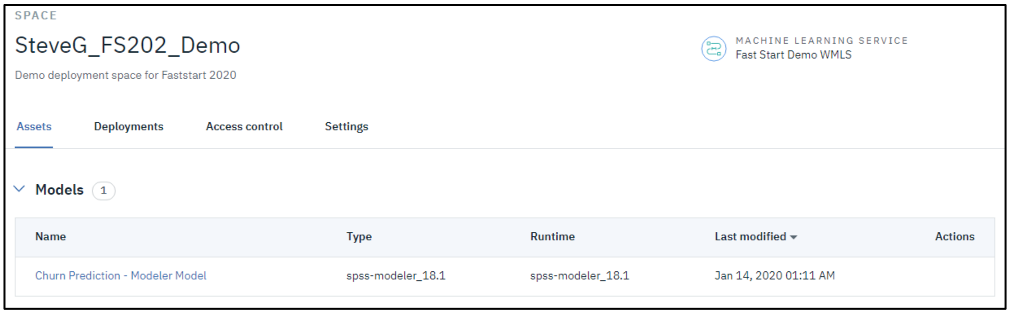


20. Click **Save** to store the model to WML Server.

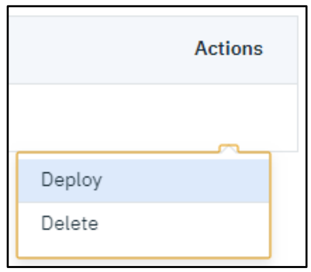


21. Return to the project view.

22. Access the deployment space by clicking the link on the project asset page,

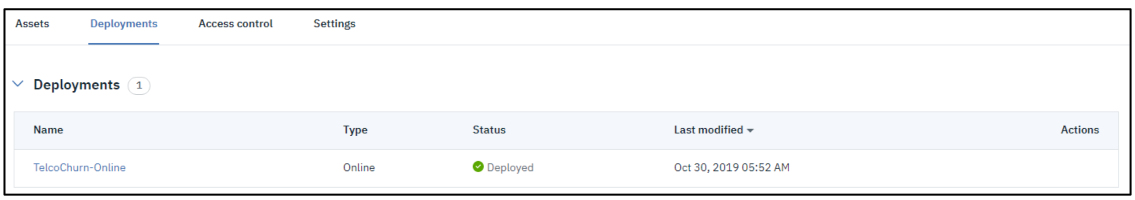


23. Click on the ellipses under the **Actions** column of the *Churn Prediction* model



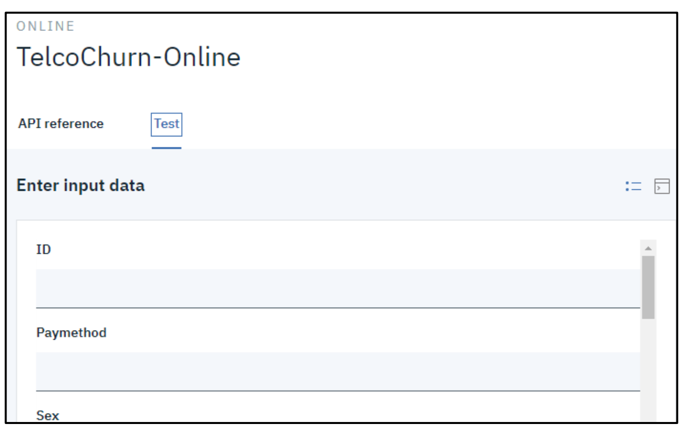
24. For the method of deployment, choose *Online*.

25. Provide a name for the deployment (e.g. TelcoChurn – Online) and click **Deploy**.



26. To test that the deployment works, click the deployment to enter the test API environment.

27. Choose the **Test** link, and enter sample data



You can use the following values for testing. You can also test with other values which you can find in the *Churn Batch Input Data.csv* file

ID	*12345*
Sex	*M*
Region	*3*
Children	*2*
Est_Income	*22358.1*
AgeGroup	*4*
Car_Owner	*N*
Status	*M*
Paymethod	*CC*
LocalBilltype	*Budget*
Customer_Segments	*High Income Families*
Trips_To_The_Website_Yr	*6*
Avg_Spend_Per_Store_Visit	*12.19*
Customer_Loyality_Code	*3*
Number_Of_Transactions_Current_Year	*8*

Scoring results will look similar to the following screenshot.

12345 – *Record ID*
Leave – *Churn Prediction from the model*
0.63492 – *Model Confidence (63.49% of customers in the node churned)*

You have finished configuring a model for online scoring.

Part 4: Model Creation using Jupyter Notebooks

In addition to Modeler flows, WSD includes a *Jupyter Notebook* development environment. In this part of the lab you will review a notebook that builds a *Churn Prediction* model.

1. From the project assets page of your *Churn Prediction* project, select **Add to project** and choose Notebook.

2. Select the **From file** option.

3. Navigate to the *Churn Prediction* subfolder of the git repo and import the notebook *Churn Prediction Notebook.jpynb*.

The notebook has been extensively documented with comments and explanations. Review the notebook to understand how to access, join, visualize and model data within the WSD python environment.

Note: ***Pay special attention to the data access section in order to ensure that the correct files are read from the project data assets.***

Summary

You have finished the WSD and WML Server lab. You can find additional documentation and tutorials in IBM Knowledge Center.

Tutorials:
https://www.ibm.com/support/knowledgecenter/SSBFT6_1.1.0/wsd/tutorials.html

WSD documentation:
https://www.ibm.com/support/knowledgecenter/SSBFT6_1.1.0/mstmap/kc_welcome.html

WML Server documentation:
https://www.ibm.com/support/knowledgecenter/SS3PWM_1.0.0/wsj/wmls/overview.html