Lab: Developing Analytics in a Visual Data Mining Workbench

November 30th, 2018

Author: Elena Lowery elowery@us.ibm.com

Table of contents

Contents

Overview

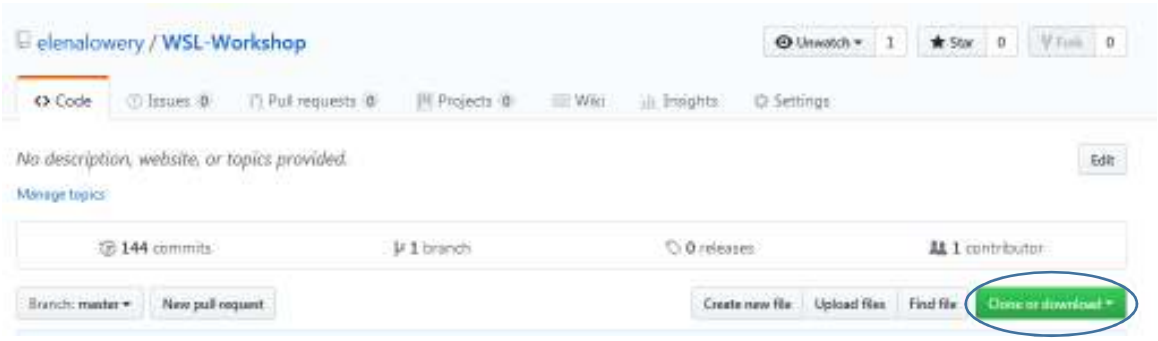
In this lab you will learn how to implement analytics in **SPSS Modeler**, a well-known visual data mining workbench which can be used in **Watson Studio.** The detailed instructions in this lab are for **Watson Studio Local** (WSL). We also provide high-level instructions for **Watson Studio Cloud**.

If you're an existing SPSS Modeler user, you will have a chance to work with the redesigned UI. If you're new to SPSS, you can still complete this lab because we provide step-by-step instructions.

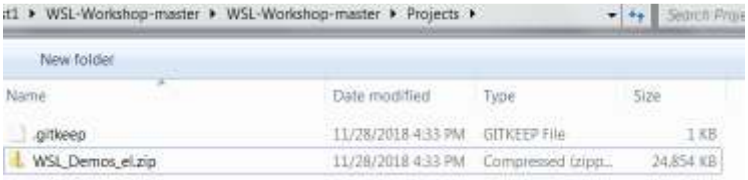
You can learn more about Modeler in WSL in official product documentation:
https://content-WSLlocal.mybluemix.net/docs/content/local-dev/spss-modeler.html

Required software, access, and files

- To complete this lab, you will need access to a **Watson Studio Local** cluster.
- You you will also need to complete the following steps to import the sample project (if you haven't done it in one of the previous labs):
    - Download and unzip this GitHub repository:
      https://github.com/elenalowery/WSL-Workshop



    - In the **Projects** directory of the unzipped file, rename *WSL_Demos.zip* to a unique name, for example, add your initials.



    - Log in to **WSL** and create a project **From File**, using the *WSL_Demos.zip* file that you just renamed.

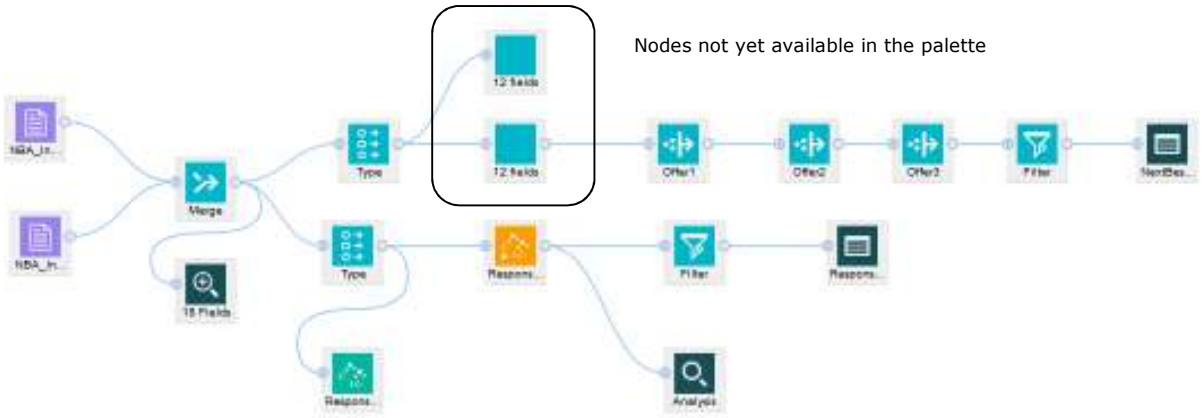
Part 1: Introduction

Notes for current SPSS Modeler users

Modeler in WSL uses the same server runtime as Modeler stand-alone (Modeler Server). The first version of Modeler in WSL does not have all the same visual nodes as Modeler stand-alone. IBM is working on implementing the majority of Modeler stand-alone capabilities to WSL.

Modeler streams in WSL are called "flows".

If a visual node is not yet available in Modeler, you can still import and run it in WSL. The node will not have an icon.



Some features, like scripting, looping, and open source integration are not yet available in Modeler/WSL. Please check with the lab instructor if you would like to see the complete list of features that are not yet available.

Modeler in WSL does not yet support all the same data sources as Modeler. See Compatibility Reports (**Reference** section) for the list of supported data sources.

Modeler in WSL uses JDBC drivers for supported data sources (see **IBM Compatibility Reports**). SQL pushback is enabled by default for all supported database data sources. At this time there is no visual indicator for SQL pushback. Use *Modeler documentation* to understand which nodes can be pushed back.

Deployment is done in **Watson Machine Learning**. At this time only *batch* deployment of SPSS models is supported in WSL.

Introduction to Modeler for new users

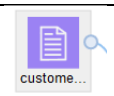
SPSS Modeler is a visual data mining workbench. Modeler can be used to complete all tasks of the analytic application development

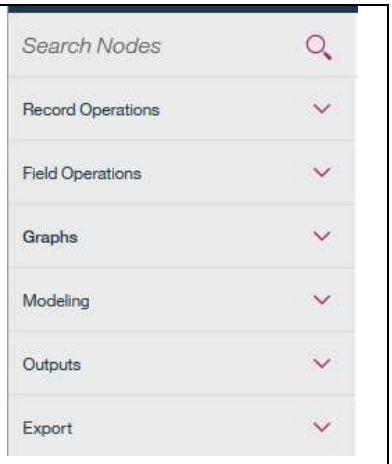
- Data understanding
- Data preparation
- Model building
- Model evaluation.

Deployment of SPSS flows is done in **Watson Machine Learning (WML)**, which we cover in the **WSL Deployment** hands-on lab.

Assets developed in Modeler are called "flows". Another frequently used term in Modeler documentation is "streams". A flow starts with one or several data sources. Using visual nodes, a user can apply different operations to data. Data "flows" from one node to another in the direction of the arrows.

Visual nodes in Modeler are color-coded and organized by type of operation: **Record Operations**, **Field Operations, Graphs, Modeling, Output,** and **Export** (data sources). Most operations are well-known functions in data preparation and analytics, such as sampling, filtering, binning, etc.

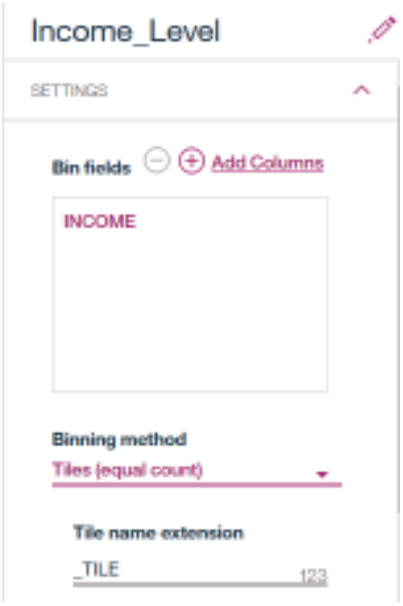
The data sources are purple	
Data preparation operations are blue	
Algorithms are green	
The models that are created based on algorithms are orange	
Different types of output (graphs, tables, external files) are black	
The nodes with a star icon are called "supernodes" - they contain several nodes. Supernodes are used for visual organization of the flow.	



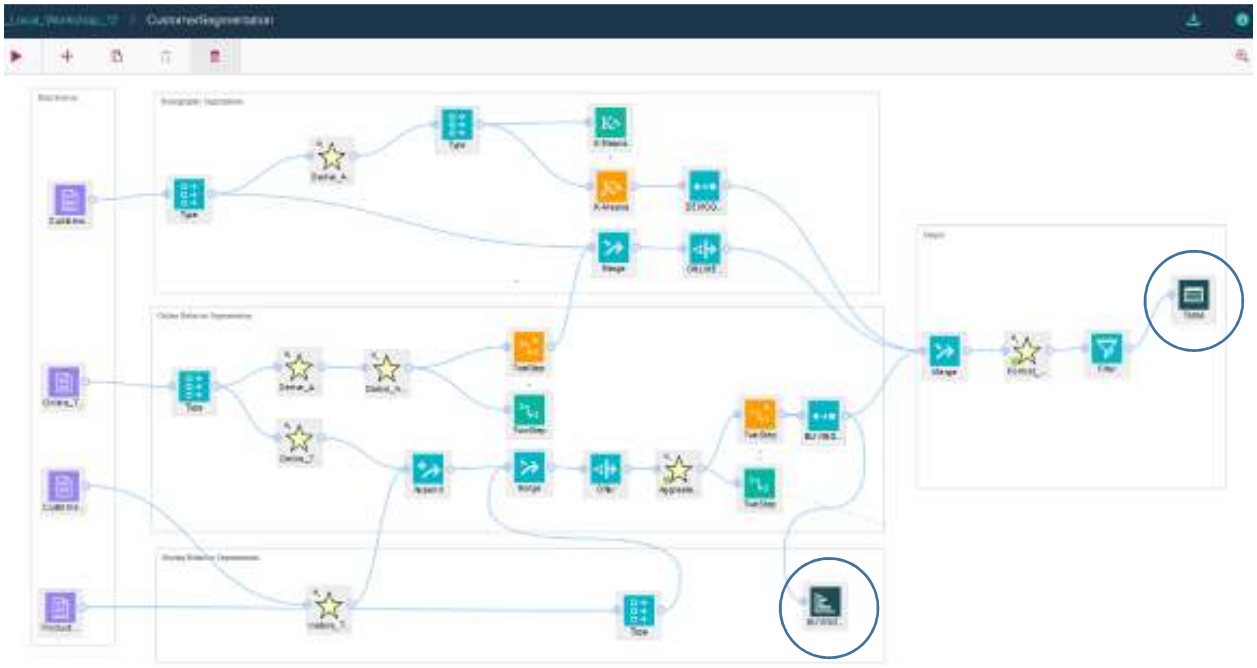
If a user needs more information about a particular node, it can be looked up in Modeler documentation. SPSS also publishes the **Algorithms Guide** that explains how machine learning algorithms are implemented in Modeler (see **Reference** for more information).

All visual nodes in Modeler have editable setting. You can bring up the **Settings** view by double clicking on the node. Nodes should be edited after they have been connected to the previous node because Modeler is aware of data definitions, types, and manipulations that happen in the flow. For example, if we use a **Derive** node to create a new field, the next node after it, a graph, can automatically detect the derived field.

Settings for a Binning node



A node that doesn't have any outgoing connections is called a "terminal node". While any node can be a terminal node, typically a terminal node is a final step in a process: it may be an export to a data source, a visualization, an evaluation node, or a table that displays output. As shown in this example, the flow can have several terminal nodes.

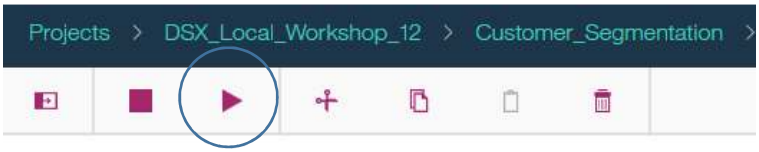


When working on the flow, we can execute different parts of the flow.

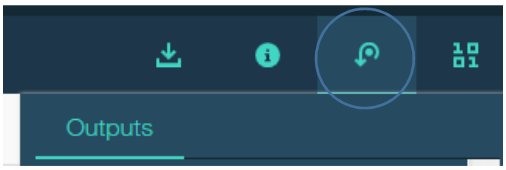
- If we click on the arrow icon in the menu bar, it will run all nodes of the flow.

- If the flow has output (visualizations, tables), they will be available in the **Output** tab of the Modeler UI.
- If the flow builds a model, it will build the model node or replace the existing one.

Run entire flow

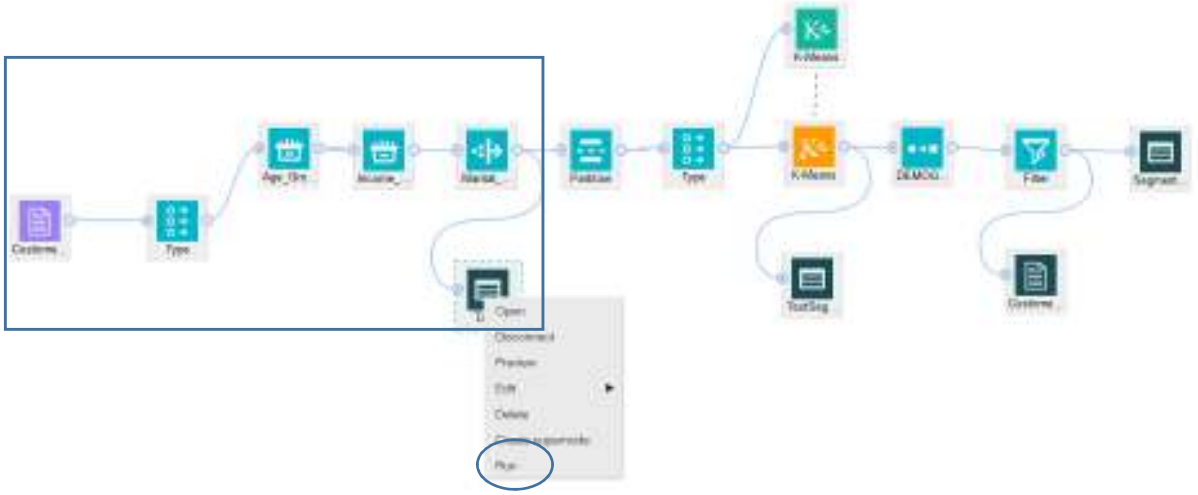


Outputs tab



When a flow has several terminal nodes, it has several "branches". We can run each branch by selecting the terminal node and running it.

Selecting Run in the node menu runs the flow branch up to the selected node.



If the flow has an **Export** node (a data source), when the entire flow or the export node branch is executed, data is written to the data source.

Use the Run button on the menu or on the selected Export node to write data to an external data source.

We will use most features described in this section in the lab.

Part 2: Review and edit a flow

In this section we will review a previously created flow, make changes, and run it.

1. Log in to a **WSL Local cluster** and open *WSL_Demos* project that you used in the previous lab.

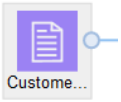
2. Switch to the **Assets** tab and click on *SimpleCustomerSegmentation* flow to open it.

   This flow performs several tasks to segment each customer into a demographic cluster. This type of segmentation can be included in a more sophisticated analytics workflow to create a customer profile. A complete customer profile can then be used to create more effective marketing campaigns.

3. First, locate the data source in the flow. If we need to add new data sources, we can select them from the data sources view. Click on the **Data Sources** icon to view available data sources.

   *Note: IBM publishes a compatibility report which shows the data sources supported by Modeler in WSL. See the **Reference** section.*

   We have 1 data source in this flow, which contains demographic information about the customer.



4. If you want to preview data in the data source, click on the data source and select **Preview**.

Preview typically loads the first 10 rows of data.

CUST_ID	REGISTRATION_ID	NAME	AGE	GENDER	MARITAL_STATUS	PROFESSION
292	MICHAEL.TOWB		33	F	M	Private Sect
305	TREVOR.KODJO		33	F	M	Professor
563	BILLY.POLO@EMAIL		33	F	M	Doctor
667	ADRIAN.DERN@EM		33	F	M	Engineer
1157	CHARLIE.HOUSE@E		33	F	M	Not employe
273	HELENA.KODJO		34	M	M	Professor

5. Next, we'll take a look at some of the data understanding features. Right mouse click on the data source, *CustomerDataSegmentation*, and select **View Data**.

   Click on the **Spreadsheet** icon on the left.

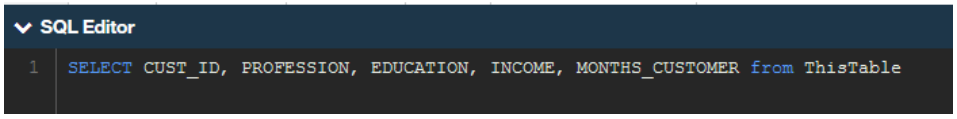
   You can sort data by clicking on column headings.

   If you would like to remove some columns from the view, click on the **SQL Editor** at the bottom of the page.

   In this case "SQL" referes to the syntax that will be used. In our example we are working with data in a flat file. This editor is used to modify the view only, it does not persist the changes.

   Try filtering out a few fieds, for example, select only the following fields. Click the arrow in the right corner to run the scipt.

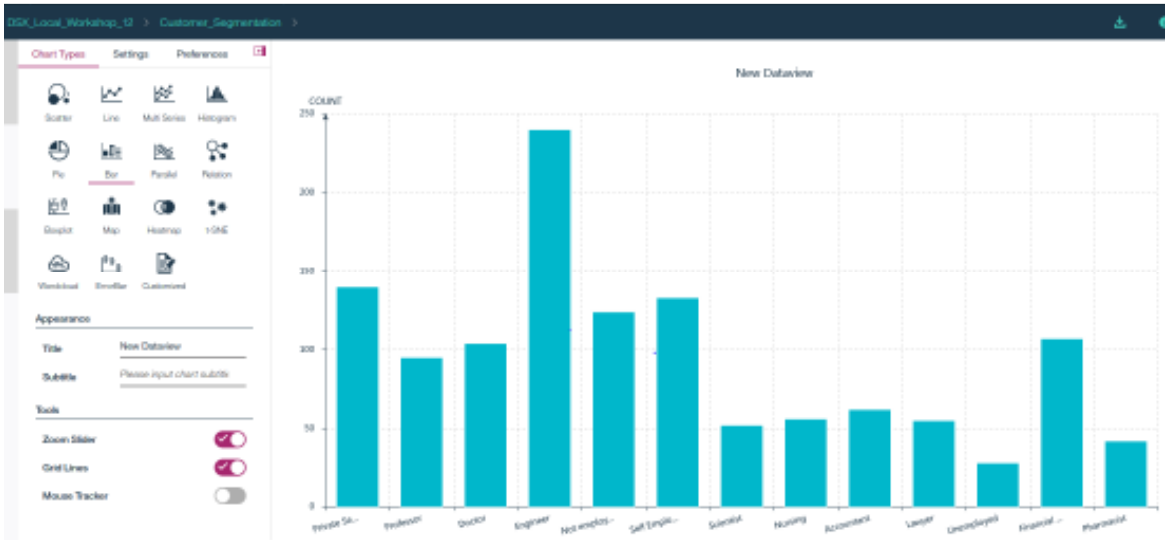
SELECT CUST_ID, PROFESSION, EDUCATION, INCOME, MONTHS_CUSTOMER FROM ThisTable

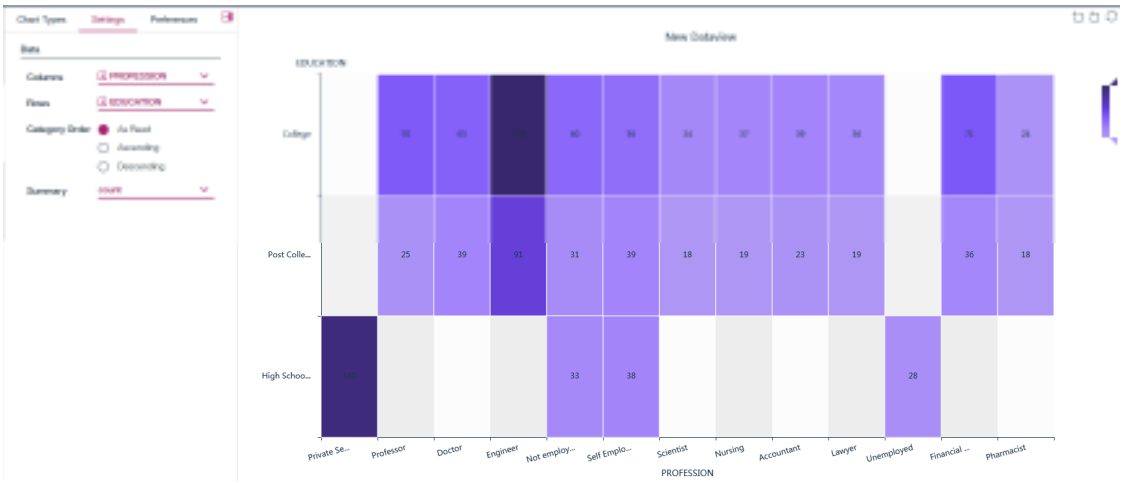


Switch to the **Data Audit** view and review statistics for each column.

Switch to the **Chart** view and try creating a few charts. For example, create

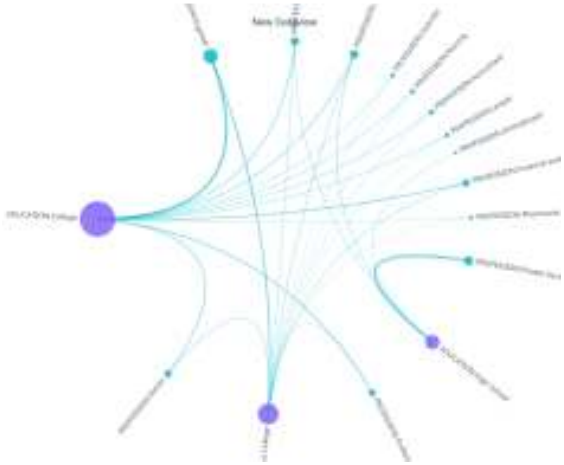
- A bar chart of profession
- A heat map of education and profession





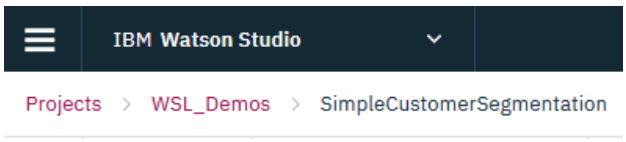
You may have noticed that not all fields are displayed for some graph types. **Modeler** automatically displays only the fields that are applicable for a specific graph type.

For example, the *Relationship* graph can only work with 2 string fields, that's why only *Education* and *Profession* are shown as available options. If you would like to create graphs for numeric fields, you can use binning or derive functions to prepare the data.

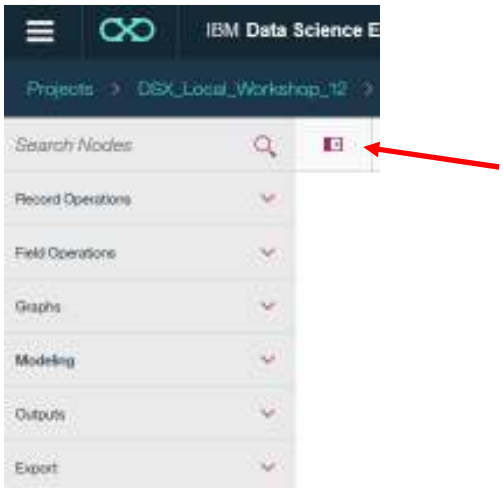


Notice that you can download and save each graph.

6. Navigate back to the flow by clicking on *SimpleCustomerSegmentation* link in WSL navigation menu bar.

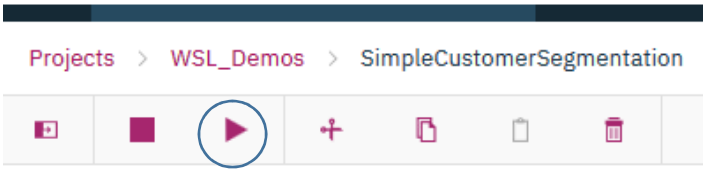


7. The rest of the visual nodes in the flow have been selected from the nodes palette. To open the palette, click on the palette arrow icon in the left corner. As we discussed in **Introduction**, the visual nodes are organized into categories based on a function that they perform.

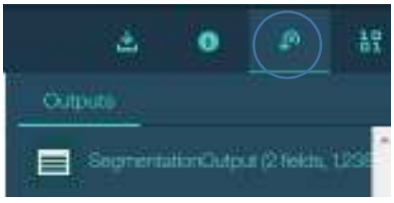


The majority of operations in the flow are data preparation. We create 1 model, a *K-means* segmentation model, and one output, a table, which shows the results of segmentation.

8. Let's run the stream, and then we'll take a closer look at the implementation. Click the arrow icon to run the entire stream.



9. The output is accessible through the **Output** view. The output table displays the segmentation results.



10. Double click on output to view it. The output shows customer id and demographic cluster (the value that was generated in the flow).

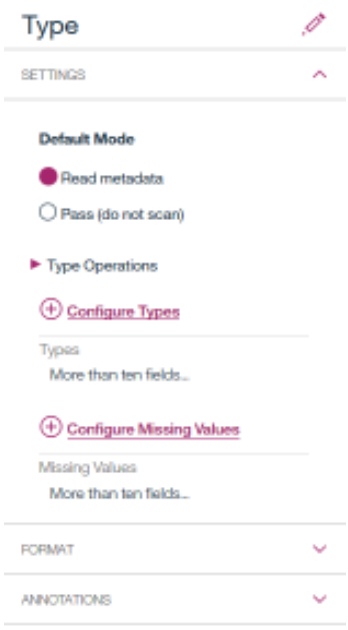
A *Table* is used to display data in Modeler. Later we will add an **Export** node which will write data into an external data source.

11. Next, let's review some of the frequently used nodes.

    The data source node is connected to a **Type** node. Since Modeler node functions can be different depending on a data type (like we've seen in the example of graphs), it's important to identify the data type before we apply any operations to data.

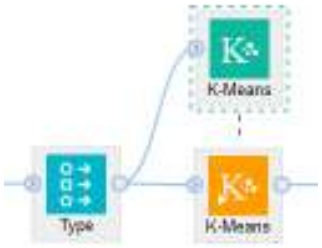
    Modeler performs "auto typing", and in most cases the default node settings don't need to be modified. However, if users want to modify this node, it can be done through the Settings view.

    Double click on the first **Type** node connected to the *CustomerDataSegmenation.csv* data source and review the settings. Notice that you can do some basic data transformation in this node (Click **Configure Types**).



    We also have a second **Type** node in the same branch, right before the *K-means* algorithm node. When the **Type** node is placed right before the algorithm, it's used to specify fields that will be used for modeling, the target fields, and fields that should be discarded.

Double click on the second **Type** node, and take a look at the roles that have been specified for modeling (Click on **Configure Types**).

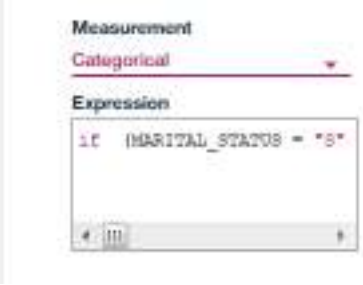


After the first **Type** node we have several data preparation operations that are common in the analytics process, such as binning, deriving new variables, and partitioning data before modeling.

If you position the mouse over the node, it will show a tooltip with the node type.

Similar to the **Type** node, you can view the settings for each node by double clicking on it or selecting **Open** from the right click menu.

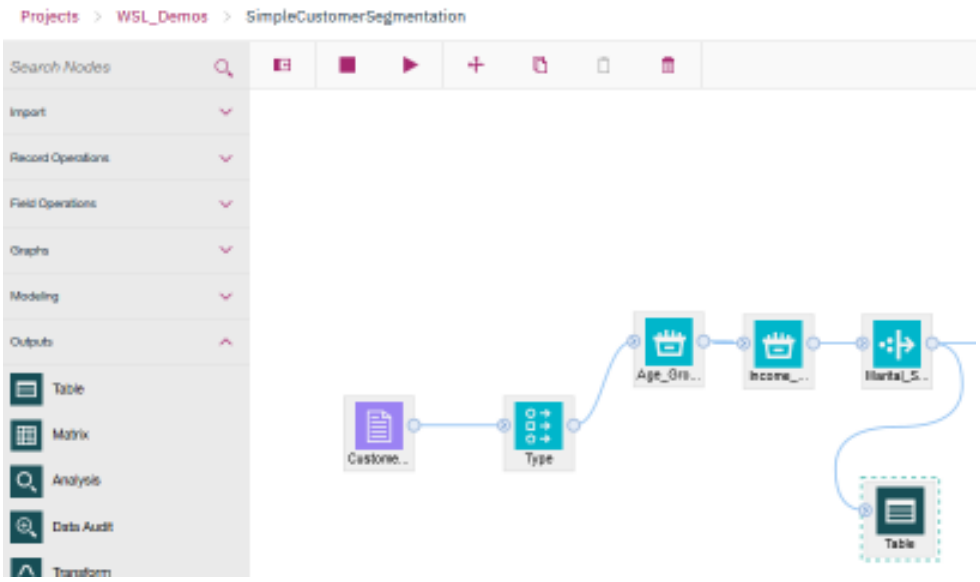
If you view the **Settings** for the *MaritalStatus_Derived* field, you'll see that it contains a formula. The formula derives the marital status if it's empty.



This formula is called an "Expression" in Modeler. In the future releases, Modeler in WSL will have an "Expression Builder", which supports building expressions by selecting fields and actions in a dialog box (i.e. you will not have to write the code). This capability is already available in Modeler desktop.

At this time, you can use documentation to look up syntax for expressions: https://www.ibm.com/support/knowledgecenter/en/SS3RA7_18.1.1/modeler_mainhelp_client_ddita/clementine/clem_function_ref.html

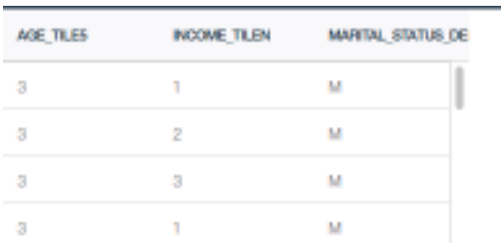
12. Attach a **Table** node (from **Output** tab) to the **Marital_Status_Derived** node.



13. Right mouse click on the **Table** node and click **Run**.

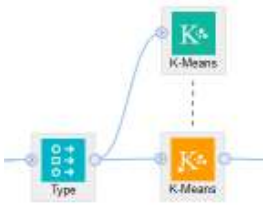
When we select **Run** from a node menu, all nodes from the source till that node are executed. Output is accessed through the same **Output** view.

Double click on the *Table*, and scroll all the way to the right. The fields that were derived by the data preparation nodes are appended at the end of the dataset. *AGE_TILE5* and *INCOME_TILEN* were created by the **Binning** nodes, and *MARTITAL_STATUS_DERIVED* was created by the **Derive** node.

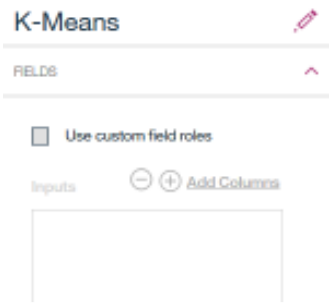


Select **SimpleCustomerSegmentation** in the menu bar to return to the model view.

14. Next, let's review the algorithm and the model nodes. The green icon is the algorithm which was selected from the **Models** palette, and the orange icon is the model that was created based on provided dataset.



Double click on the green **K-means** icon and review the properties. The **Fields** are empty because the algorithm is using field "roles" (input, target, none) from the **Type** node in front of it. This is typical implementation in flows – to specify field roles in a separate **Type** node.

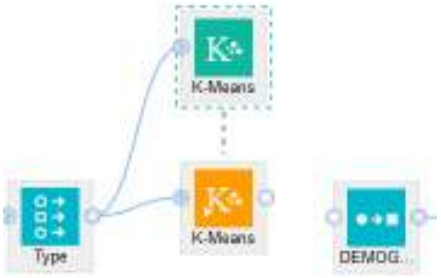


If you wish, review the **Type** node to understand which fields are used for training this model (*Hint: double click on **Type** node, then select **Configure Types***).

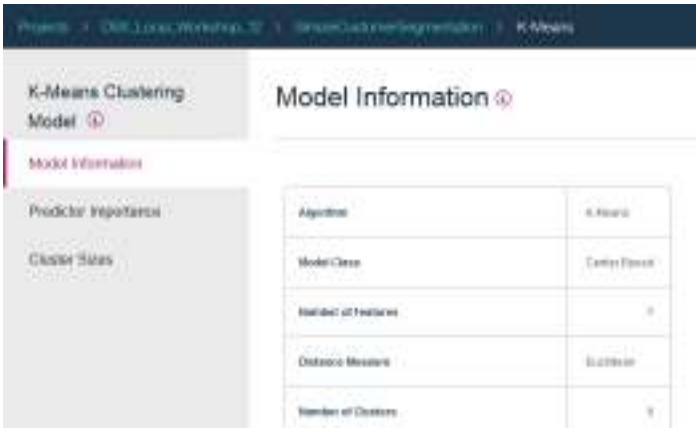
The rest of the settings in the algorithm were provided by default. An experienced data scientist can use **Build Options and Expert Options** for each model type, but in most cases the default values provided by Modeler don't need to be changed.

Let's delete the generated model and build a new one. Select the orange **K-means** icon and delete it.

Right mouse click on **K-means algorithm** (green icon), and select **Run**. This generates the *K-means* model.



15. Right-click on the generated model and select **View Model** to review model details. This information helps data scientists to understand the model.

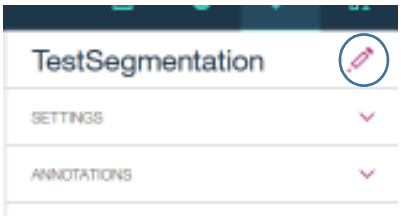


Return to the flow view.

16. Add a table to the **K-means** model node.

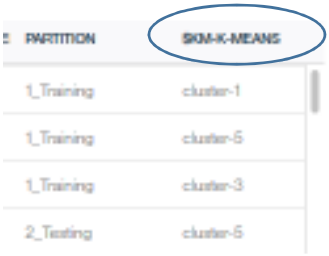
*Hint: open the Palette, then open Outputs and drag a table icon onto the canvas. T*hen link the table to the model.

Double click on the table and name and in the **Settings** view change the name to *TestSegmentation*. If we don't rename the table, it may be confusing to find it in the **Output** view. Make sure to save the changes for the Table node.



Run the flow up to the table by selecting **Run** from the *TestSegmentation* table right click menu. View the output.

The model generated the cluster field, which was appended to the end of the data source.



The cluster information without explanation may not be useful to a marketing professional, that's why we need to add a *Reclassify* node, to give the clusters an easily understandable name.

Note: Modeler desktop segmentation algorithms provide an easy way to understand the composition of the cluster, and this capability is currently being implemented in WSL. We implemented a separate Modeler stream, ClusterUnderstanding, to explain how it can be done at this time. Please check with your instructor if you would like to learn more about the approach we used.

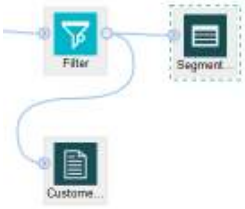
17. Connect the generated model to the **Reclassify** node next to it. The **Reclassify** node assigns a more meaningful name to a cluster.

Double click on the **Reclassify** node to review how clusters are renamed.

18. The next node is the **Filter** node. This node simply removes the fields that we don't need in the final output. Double click on the **Filter** node and review its properties.

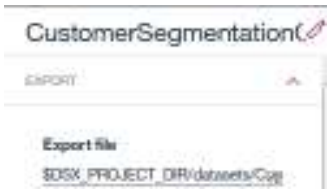
19. At this time the final output of the flow is displayed in a **Table** node, which is internal to Modeler. Let's add an **Export** node, which will write output to a file or a database datasource.

    Connect a **Flat File** node (from the **Export** tab in the palette) to the **Filter** node.



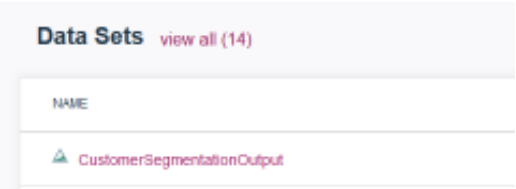
    Edit the **Flat File** node to give it a name (change in properties) – for example, *CustomerSegmentationOutput.csv*. It's important to add .csv extension to enable preview and download capabilities in WSL projects.

    Note: make sure not to delete the path before the file name.



20. Run the entire stream by selecting the **Run** icon from the main menu bar. Navigate to the WSL **Project** view.

    The flow generated the output file which you can preview and export (use the vertical ellipses button).

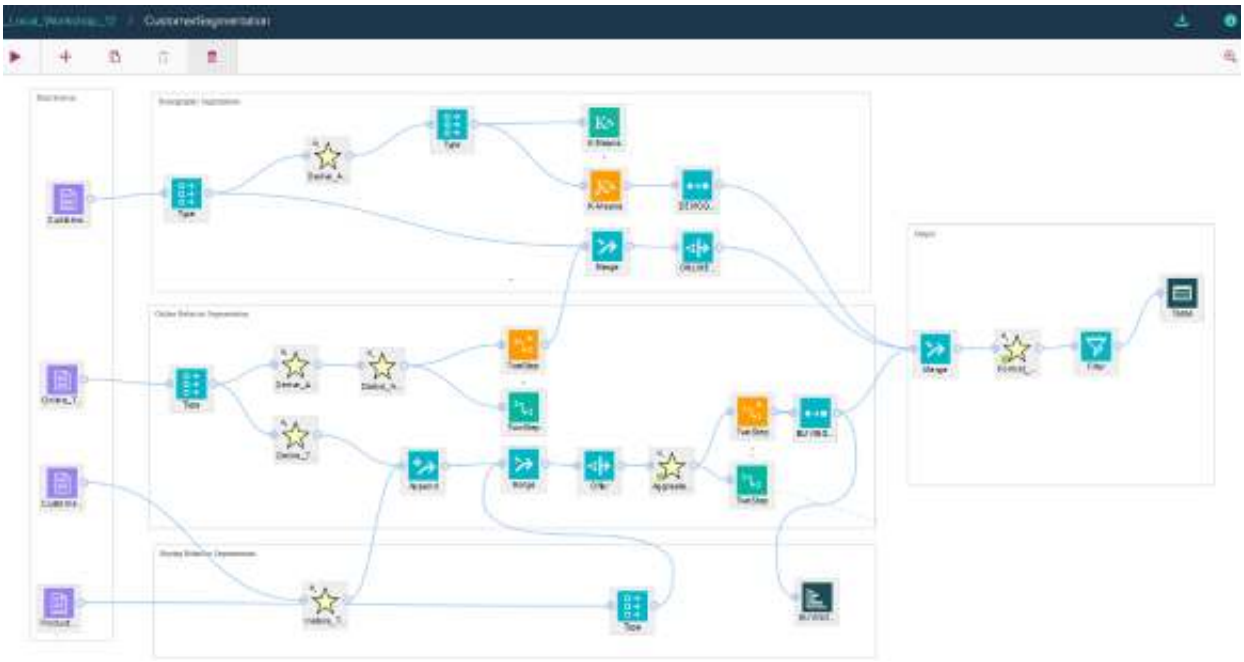


We have finished reviewing a simple Modeler flow. Now let's review a more complicated one.

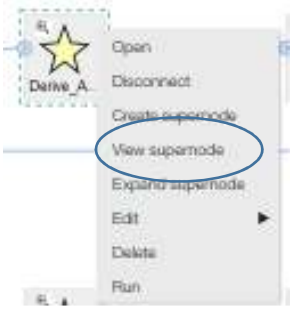
21. Switch to the **Assets** tab and click on *CustomerSegmentation* flow to open it.

    This flow performs several tasks to derive the following information for each customer: Customer Lifetime Value (LTV), buyer category, demographic cluster, online behavior cluster, and buying behavior cluster. This information can help companies run more effective marketing campaigns.

    The top part of the flow performs demographic segmentation, which we already reviewed.



The yellow icons with a star are called "*supernodes*". They are used for visual organization in a flow. Supernodes contain several other nodes. Click on one of the supernodes and select **View Supernode**.



Take time to review this flow and let the instructor know if you have questions about nodes or the analytics implementation.

Try running this flow and review the outputs such as *SegmentationOutput* table. In addition to the demographic cluster, which we generated in the demographic segmentation flow, the flow creates additional fields that create a more complete customer profile.

CUST_ID	REGISTRATION_ID	LTV	DEMOGRAPHIC_CLUS	ONLINE_BEHAVIOR_CI	BUYING_BEHAVIOR_C
		LOW VALUE	MIDDLE INCOME SI	PRODUCT VIEWER	INSTORE MEDIU
0.000		LOW VALUE	MIDDLE INCOME S(	VISITOR	INSTORE MEDIU
1.000	50524	HIGH VALUE	MIDDLE INCOME S(	VISITOR	INSTORE LOW VA
2.000	226666	LOW VALUE	MARRIED WOMEN	VISITOR	INSTORE MEDIU
3.000	254376	MEDIUM VALUE	MIDDLE INCOME S(	VISITOR	ONLINE MEDIUM
4.000	288205	MEDIUM VALUE	HIGH INCOME FAMI	VISITOR	INSTORE LOW VA
5.000	660891	LOW VALUE	MARRIED MEN	VISITOR	INSTORE MEDIU

Part 3: Create a new flow

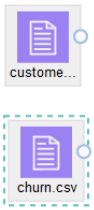
In this section you will create a Modeler flow in WSL/Modeler UI. The process is the same as creating a Modeler stream in **SPSS Modeler** desktop, the only differences are in the UI look and feel.

If you have worked on other hands-on labs in the WSL workshop – this Modeler flow implements the same use case as Telco Churn Jupyter and Zeppelin notebooks (available in the *WSL_Demos* project).

1. On the **Assets** tab scroll down to **SPSS Modeler flows** and click **add flow**.

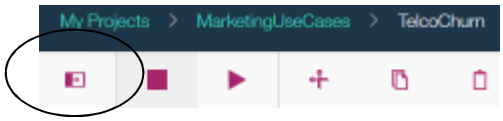
2. Provide a unique stream name (for example, add your initials) *TelcoChurn_el*. Click **Create**.

3. Drag and drop *customer* and *churn* data sets to the canvas.

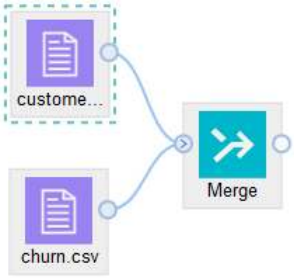


4. Right mouse click on the data sources in the canvas and select **Preview** to view the data. *Customer* dataset contains demographic and service usage information and *churn* dataset contains historical data showing if the customers has churned.

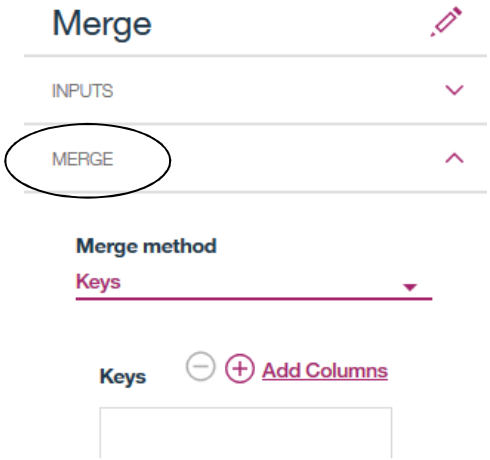
5. Click on the **Palette** icon and expand **Record Operations.**



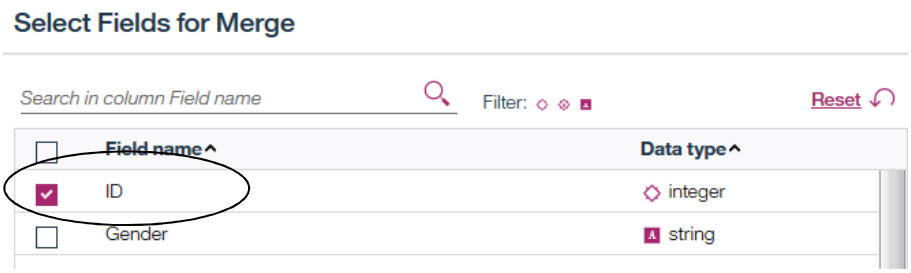
6. Add the **Merge** node, then connect the *customer* and the *churn* data sources to it.



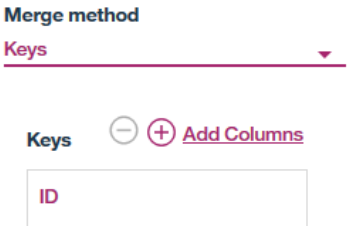
7. Double click on the **Merge** node. Select *Keys* as the **Merge method**.



8. Click **Add Columns** and select the *ID* field. Click **OK** to return to the previous screen.

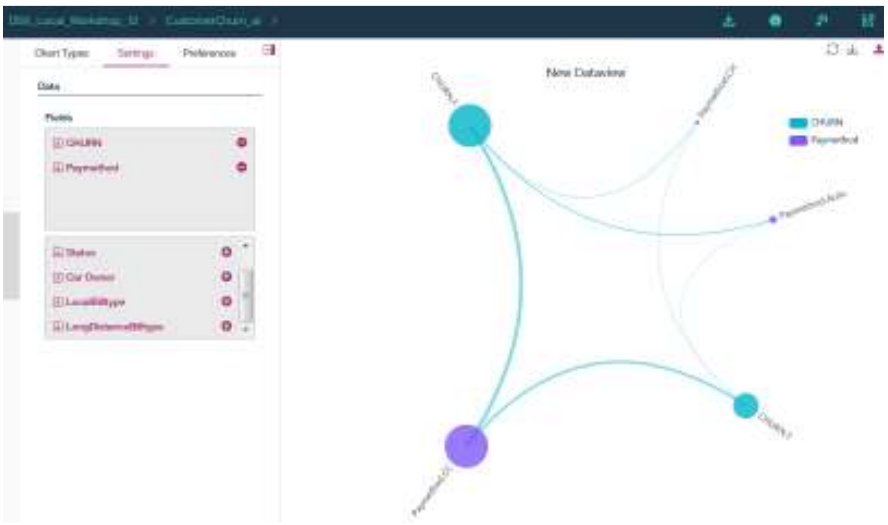


9. Now the **Merge** screen looks like the following screenshot. Click **Save**.

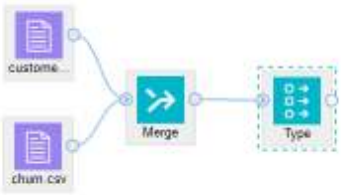


Next, we can perform several data understanding tasks by selecting **View Data** from the right click menu of the **Merge** node.

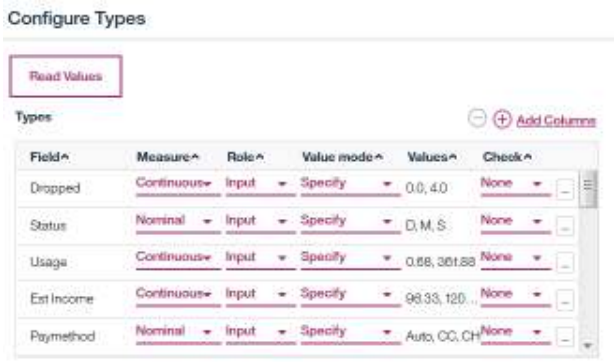
Try several different steps, as we've done in the previous section. For example, you can create a relationship graph for churn and payment method. This graph shows that a small percentage of customers who use autopay have churned.



10. Next, we are going to build a model for predicting churn. Add a **Type** node from the **Field operations** and connect it to the **Merge** node.



11. Double click on the **Type** node and click **Configure Types**. In the **Configure Types** screen click **Read Values**.

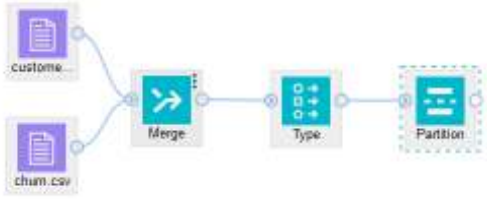


12. Sort fields by name (so that it's easier to find them). Change the **Role** of *churn* to **target** and *ID* to **Record ID**.

    This means that all fields will be used as input to modeling, with the exception of the *ID* field. *Churn* is the field we want to predict so set its role to **Target**.

    Click **OK**, then **Save** to save the settings in the **Type** field.

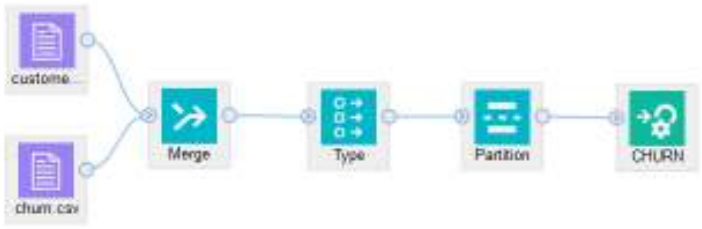
13. From the **Field Operations** add the **Partition** node and set the **Training Partition** to *70* and the **Testing Partition** to *30*. Save the changes in the node.



14. Add the **Autoclassifier** node from the **Modeling** tab. **Autoclassifier** will automatically create several classification models and show model accuracy for the best models.

    Modeler documentation explains which classifier models are built, and why some models are discarded (see **Reference**).

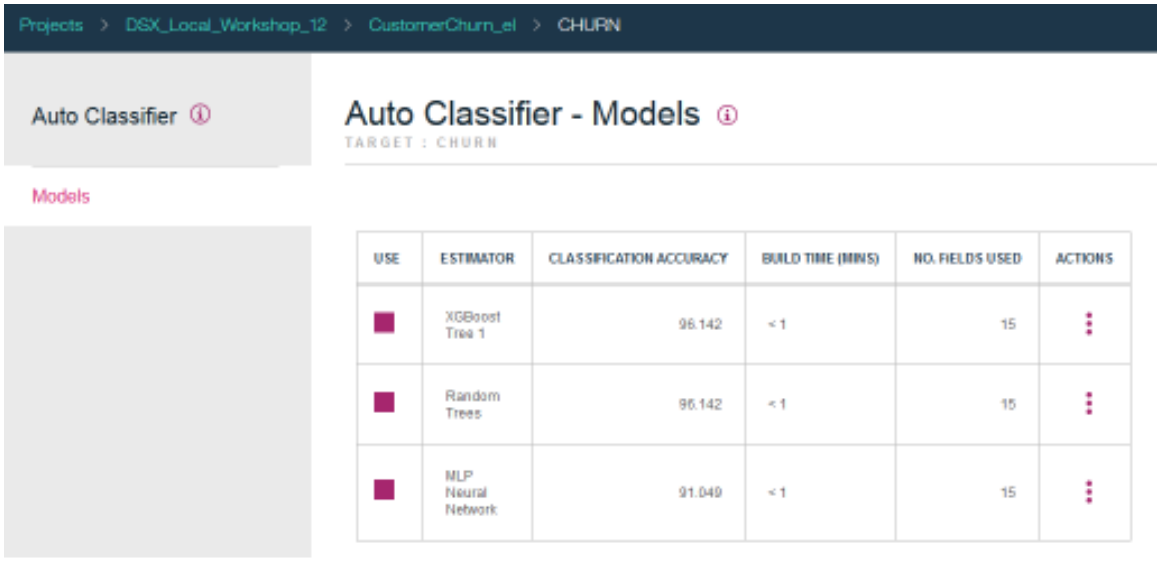
The **Autoclassifier** node is named *Churn* because we specified churn as the target field in the **Type** node.



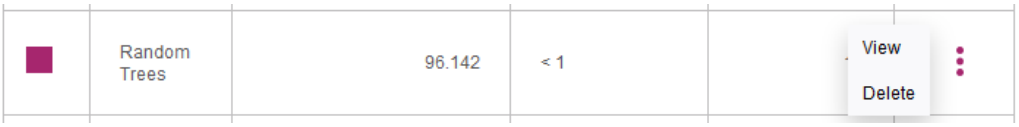
15. Right click on the **Autoclassifer** node (*CHURN*) and select **Run**.

16. Model building will take a few minutes. When model building is done, you'll see the model node (orange icon on the canvas).

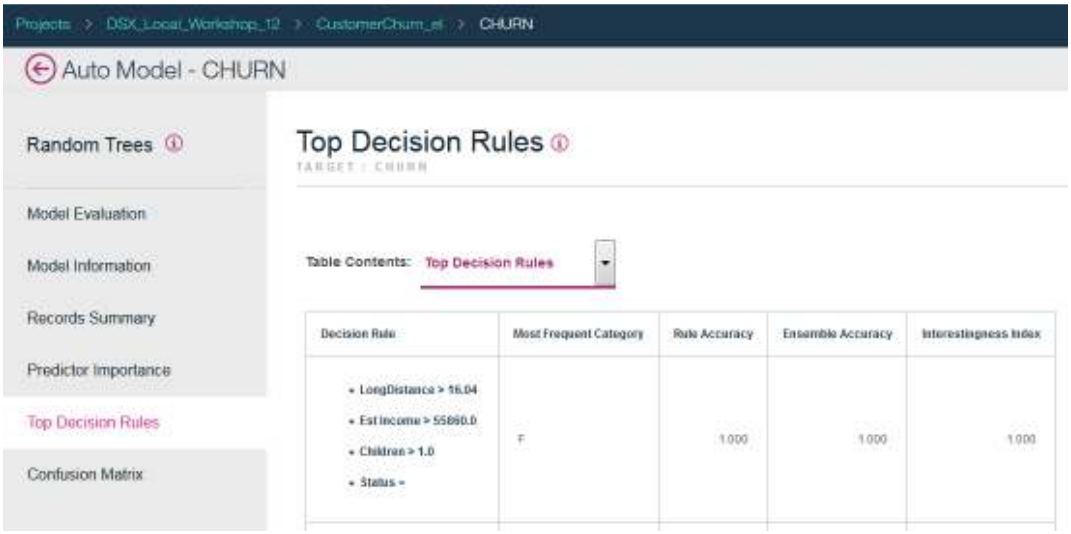
    Right-click on the generated model and select **View Model**. The top three models generated are shown in the screenshot.



17. Click on ellipses next to *Random Trees* model and select **View**.



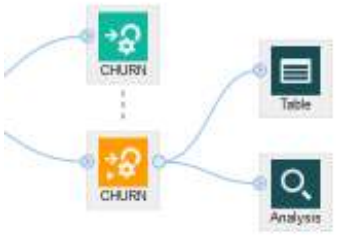
Review model information. For example, the **Top Decision Rules** view is useful for understanding the factors that lead to customer churn.



By default ensemble scoring will be used. Some ensemble scoring settings can be changed in model properties. As you've seen in the model details view, models can be deleted (removed from the ensemble).

In some cases data scientists use the "auto" modeling nodes to determine the best model, and then they use that individual model from the models palette.

18. Attach **Table** and **Analysis** nodes to the model, and run each one by selecting **Run** from the right click menu.



Note: if you select Run from the main menu, Modeler will run the entire stream, which will rebuild the model.

Review the output. The table shows the predictions that were generated by the model (scroll to the right).

PARTITION	$XF-CHURN	$XFC-CHURN
1_Training	T	0.984
1_Training	F	0.819
1_Training	F	0.782
2_Testing	F	0.788
1_Training	F	0.974
1_Training	F	0.983

Model analysis output

Projects > DSX_Local_Workshop_12 > CustomerChurn_el > Analysis of [CHURN]

Results for output field CHURN
Comparing $XF-CHURN with CHURN

'Partition'	1_Training		2_Testing	
Correct	1,393	98.24%	625	96.45%
Wrong	25	1.76%	23	3.55%
Total	1,418		648	

19. Finally, add a **Filter** and **Export (Flat File)** nodes.

- In the **Filter** node remove all fields with the exception of ID, predicted churn value *($XF-CHURN*), and confidence in the prediction *($XFC-CHURN*),
- In the **Export** node specify a .csv file as an export.

20. Run the file export branch: right mouse click on the flat file output node and select **Run**.

21. Navigate to the **Project** view and review the generated file.

You have finished building a Modeler flow in WSL.

Working with SPSS Modeler in Watson Studio Cloud

The same Modeler flows (and instructions) can be used in **Watson Studio Cloud**. At this time (Q1 of 2019) the version of Modeler in Watson Studio Cloud is a newer version compared to WSL. You may see additional functions (for example, more algorithms and *Expression Builder*) in Watson Studio Cloud.

Complete the following steps to test Modeler flows:

1. Import Modeler flows into your Watson Studio project from the *SPSS Modeler/streams* directory of the *WSL_Demos* git repo that you downloaded earlier in the lab.
2. Import data files from the SPSS Modeler/data directory of the *WSL_Demos* git repo.
3. When opening the Modeler flow in Watson Studio, you will be prompted to migrated input data sources – migrate them to the corresponding csv files.

Summary

In this lab you learned how to work with SPSS Modeler in Watson Studio. SPSS flows can be deployed for batch scoring in Watson Studio Local (WML).

Reference

Modeler in WSL documentation: *https://content-WSLlocal.mybluemix.net/docs/content/local-dev/spss-modeler.html*

Modeler documentation:
https://www.ibm.com/support/knowledgecenter/en/SS3RA7_18.1.1/modeler.kc.doc/clementine/knowledge_center/product_landing.html

Algorithms Guide:
ftp://public.dhe.ibm.com/software/analytics/spss/documentation/modeler/18.1.1/en/AlgorithmsGuide.pdf

Compatibility Reports:
http://publib.boulder.ibm.com/infocenter/prodguid/v1r0/clarity/prereqsForProduct.html