Data Science without a PhD featuring IBM SPSS Modeler

Lab Exercise Guide

Jos den Ronden

IBM - Data and AI

jdenronden@nl.ibm.com

Think 2021

Notices and disclaimers

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

This document is current as of the initial date of publication and may be changed by IBM at any time. Not all offerings are available in every country in which IBM operates.

Information in these presentations (including information relating to products that have not yet been announced by IBM) has been reviewed for accuracy as of the date of initial publication and could include unintentional technical or typographical errors. IBM shall have no responsibility to update this information.

This document is distributed "as is" without any warranty, either express or implied. In no event, shall IBM be liable for any damage arising from the use of this information, including but not limited to, loss of data, business interruption, loss of profit or loss of opportunity. IBM products and services are warranted per the terms and conditions of the agreements under which they are provided. The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

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

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

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

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

Workshops, sessions and associated materials may have been prepared by independent session speakers, and do not necessarily reflect the views of IBM. All materials and discussions are provided for informational purposes only, and are neither intended to, nor shall constitute legal or other guidance or advice to any individual participant or their specific situation.

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

Notices and disclaimers (Continued)

Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products. IBM does not warrant the quality of any third-party products, or the ability of any such third-party products to interoperate with IBM's products. **IBM expressly disclaims all warranties, expressed or implied, including but not limited to, the implied warranties of merchantability and fitness for a purpose.**

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

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

Table of Contents

1 Introduction to Data Science without a PhD

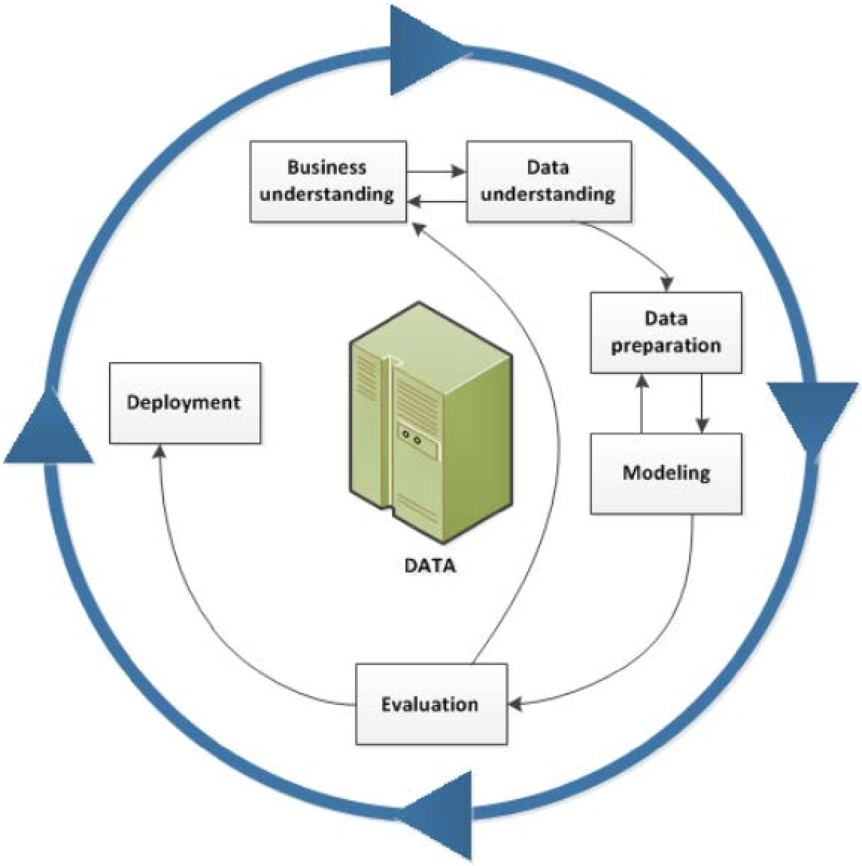
In this section, we will provide context to the lab.

1.1 Examples of business questions answered in data science projects

- Building recommendations systems, such as:
  - A music streaming service recommending certain songs or albums
  - A film streaming service recommending certain series or movies
- In banking, prevent fraudulent credit-card activity, or detect it in an earlier stage
- In banking, identify those at risk of not paying back a loan
- In insurance, prototype claims as "can pass" or "need closer inspection"
- In insurance, predict severe weather events
- In telecommunications, identify customers at risk of cancelling their subscription ("churn")
- In healthcare, reduce the incidence of a heart attack among those with a cardiac disease
- In real estate, assess the value of properties for tax purposes
- In manufacturing, identify machines or machine parts which should be replaced before they defect, thus preventing machine failures
- Optimize supply chains
- Extract sentiments from texts
- Recognize hand-written characters
- Identify spam in email
- Tag objects in images
- In call centers, decrease the time to resolution by a chatbot
- Identify objects in the environment of a self-driving car

1.2 The Cross Industry Process Model for Data Mining (CRISP-DM)

- A data science project can become complicated quickly
- To stay on track, it helps to have a roadmap or "methodology", such as CRISP-DM



1: Understand the business: Formulate the problem. For example: for a telecommunications (telco) firm, the goal might be to increase customer satisfaction, or to minimize the drop in revenues by customer churn.

2: Understand the data: Catalog the available data sources. For example, for the telco firm: you collect data on demographics, usage, logs of the call center, social media. You need to decide what data to use (or there, for example, legal restrictions?) and get thoroughly acquainted with the data.

An example of tabular data ("records" and "fields" in IBM SPSS Modeler):

ID	AGE	BUNDLE	PRODUCT CODE	USAGE (MB)	CHURN
1	18	CAT20	4.1	100	No
2		asad40	4.8	40	No
3	34	ASAD40	7.6		No
4	143	CAT30	1.1	12000	Yes
…	…	…	…		..

A field's **storage** is either:

- string (BUNDLE, CHURN), or
- numeric (ID, AGE, PRODUCT CODE, USAGE)

One of the most important concepts in analyses (and building machine learning models!) is the **measurement level** of a field. A field's measurement level defines the nature of the field's values.

- ID is a numeric field but should be excluded from all analyses. Set the measurement level of such a field to **typeless**. Typeless fields are excluded from analyses in Modeler.
- PRODUCT CODE is numeric but it makes no sense to compute its mean because its values are just product codes. Therefore, set the measurement level of such a field to **categorical**. Setting its measurement level to categorical ensures that the field will be handled as storing categories in all analyses.
- AGE is a numeric field and it is valid to compute its mean. The measurement level of such a field should be **continuous**.

Thus, storage and measurement level are different concepts: a numeric field is not necessarily continuous (such as ID, PRODUCT CODE).

BASIC RULE OF ANALYZING DATA (including modeling):

- ***If the fields' measurement levels are not correct, analyses (including machine learning models) will not be meaningful***
- ***If the fields' measurement levels are all correct, analyses (including machine learning models) will be meaningful (although maybe not optimal)***

3: Prepare the data for modeling: Fix data errors, replace missing values, and replace outlier values. This is also known as **data wrangling** in data science. Also, new fields can be derived in the hope to build better models later, which is **feature engineering**. Also, if you have many fields, you could try to select only the important fields, **feature selection**.

4: Build machine learning models: learn patterns from the data. There are various types of machine learning models. Machine learning models that predict a certain field are referred to as **supervised** models. The field that is predicted is referred to as **target**. In data science the target field is also named **label**. The fields that predict the target are referred to as **features** (and hence "feature engineering" and "feature selection" in the previous stage). In supervised models, the focus is on accuracy rather than explainability. And, in practice, "black box" models are the most accurate.

5: Evaluation: Once you have a model in place, you can predict the target and compare it to the actual behavior. For example, your model might predict that a certain customer will cancel, and that customer did actually cancel, which makes a correct prediction. Comparing the predicted values to the actual values will let you assess the **accuracy** of the model, such as the percentage of correct predictions in our example. In evaluating a model, you might also want to see if the model is fair and can be explained. Fairness and explainability of machine learning models has received a lot of attention the recent years.

ID	...	CHURN	FROM THE MODEL: PREDICTED VALUE FOR CHURN
1	...	No	No
2	...	No	No
3	...	*No*	*Yes*
4	...	Yes	Yes
...	...		..

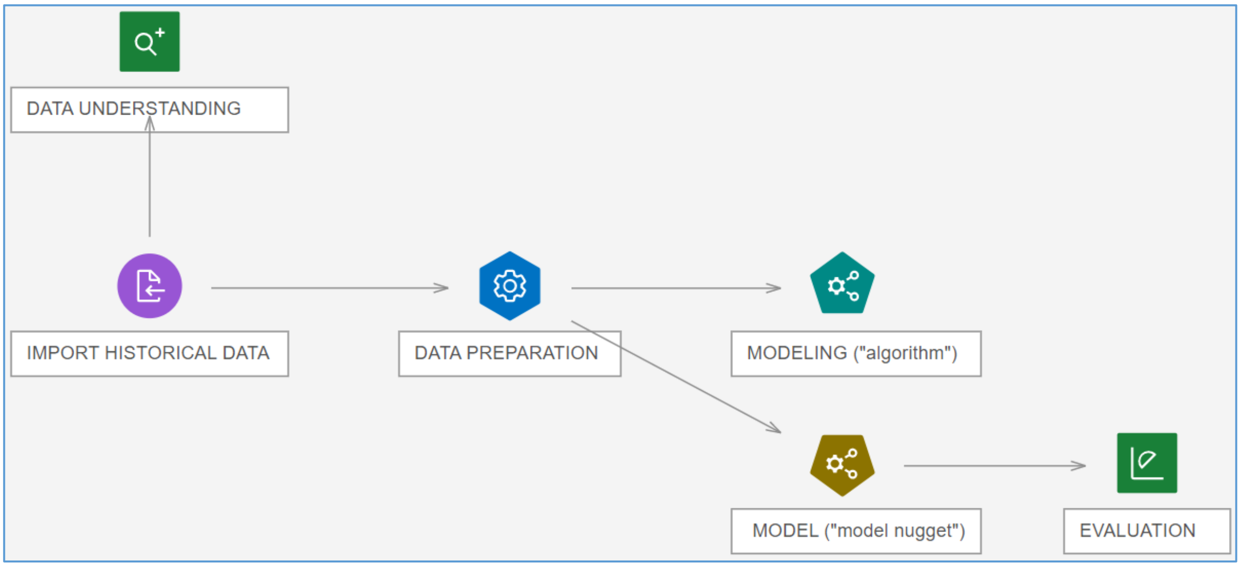
Three out of the four first records are predicted correctly by the supervised machine learning model.

6: Deployment: In this stage, the model is used to predict the behavior of new data to new data. This can be as simple as generating a report or as complex as having an app up-and-running that incorporates the model to guide customers in their decisions.

1.3 Introduction to IBM SPSS Modeler

In Modeler, you will create a "flow" to analyze your data. A flow will reflect the CRISP-DM stages, from data infusion to deployment.

The next figure shows a conceptual overview of building and evaluating a machine learning model in Modeler, along the lines of the CRISP-DM stages.



Data flows from node to node, where each node represents an operation on the data. The arrows point in the direction of the data flow. We say that "node B is **downstream** from node A", if data flows from A to B.

The above flow builds and evaluates a supervised machine learning model (supervised: a model is built to predict a certain field).

Notice the difference between a machine learning **algorithm** and a machine learning **model**:

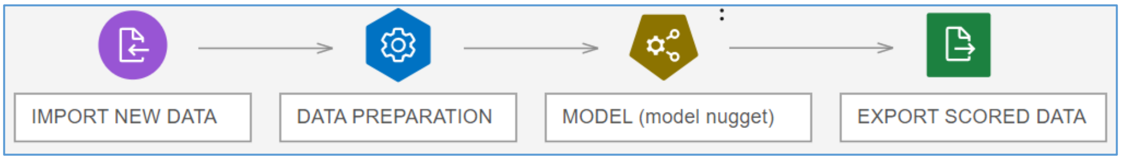
- An **algorithm** is a **computational procedure** to find patterns in the data. For example, a decision tree algorithm is optimized to find decision rules.
- A **model** is the **outcome of running an algorithm**. So, in the example of decision trees, a model stores the decision rules found by the algorithm. For example, a decision trees algorithm may have found a rule that those younger than 25 with usage less than 100 Mb have a 0.9 chance to cancel their subscription. The model will store that rule.

> In Modeler, the model is referred to as **model nugget**. Note that the shape of the model nugget is the same as shape of the 180 degrees rotated algorithm node.

In data science, the entire flow is also referred to as a **pipeline**.

Having built a model on historical data, and assuming that the model is satisfactory, you can apply the model to new data ("score data").

The following figure shows a flow to score new data.



First, import the new data. Then, apply the same transformations as you had for the historical data. Next, input the transformed data to the model. This will add a prediction for each customer to the data. You will then have a data set that stores the scored data, and you can export that data to a file (among others).

1.4 Summary of important concepts

- **CRISP-DM**: a roadmap for your data science project
- **Measurement level**: defines the nature of a field's values; can be categorical, continuous, or typeless (within Modeler, more measurement levels are distinguished but they are not relevant to this course)
  - **Categorical**: a field whose values represent categories
  - **Continuous**: a field whose values represent quantitative information
  - **Typeless**: a field that has to be excluded from all analyses
- **Basic rule of analyzing data**: ensure that the fields' measurement levels are correct to ensure that your analysis makes sense
- **Data wrangling**: preparing data for modeling
- **Machine learning algorithm**: a computational procedure to find patterns in the data
- **Machine learning model**: stores the patterns found by a machine learning algorithm
- **Model nugget**: in Modeler, the node that represents the model
- **Supervised machine learning models**: models that predict a certain field
- **Target**: the field that is predicted in a supervised model
- **Label**: synonym for "target"; other synonyms are "outcome", "response", "dependent"
- **Inputs**: the fields that predict the target
- **Features**: synonym for "inputs"; also referred to as "independents"
- **Accuracy**: in a supervised model, the degree of agreement between actual value of the target (label) and the predicted value of the target (label).

1.5 About the lab

Structure of the lab: In the first workshop you will get familiar with IBM SPSS Modeler. In the second workshop you will build a model and apply it to new data. But, it will be a quick and dirty analysis and many things can be improved. Those improvements will be the topics of the remaining workshops.

About the instructions in this lab: Actions to be performed by the user are formatted as numbered paragraphs, and interface elements in those steps are bolded. Comments or explanations to the steps are not numbered and do not contain bolded terms. Thus, if a paragraph is not a numbered step, please do not perform an action.

To complete all the workshops in this lab successfully, you will need approximately **two hours**.

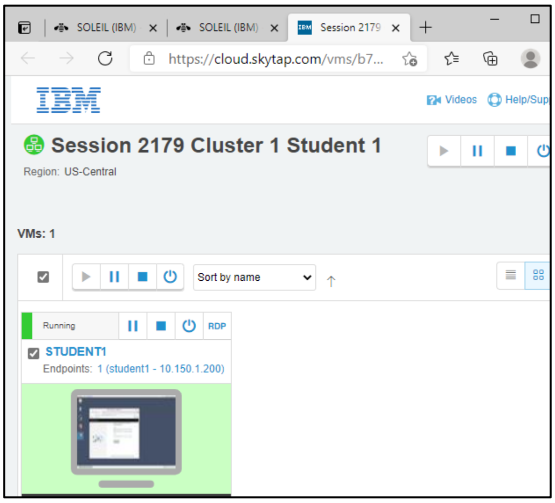
2 Create an analytics project and get familiar with IBM SPSS Modeler

In this workshop, you will get familiar with Modeler on Cloud Pak for Data. You will need to get into the virtual environment and log on to Cloud Pak for Data.

Note about the instructions in this lab: Actions to be performed by the user are formatted as numbered paragraphs, and interface elements in those steps are **bolded**. Comments or explanations to the steps are <u>not</u> numbered and do <u>not</u> contain bolded terms. Thus, if a paragraph is not a numbered step, please do not perform any action.

2.1 Start the virtual environment and log on to Cloud Pak for Data.

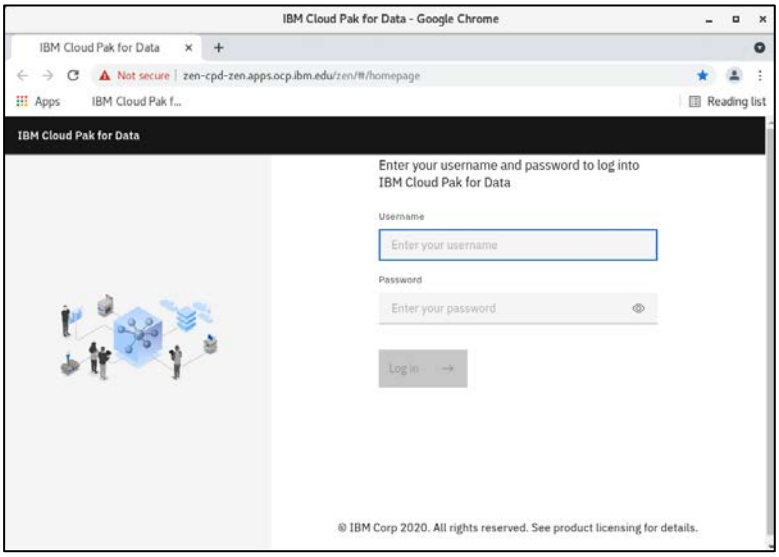
After launching the lab associated with your student id, the results appear similar to the following (shown here for Student 1):



A virtual environment is provisioned for you.

1. Click the **Virtual Machine** to access the virtual machine.

The results appear similar to the following:



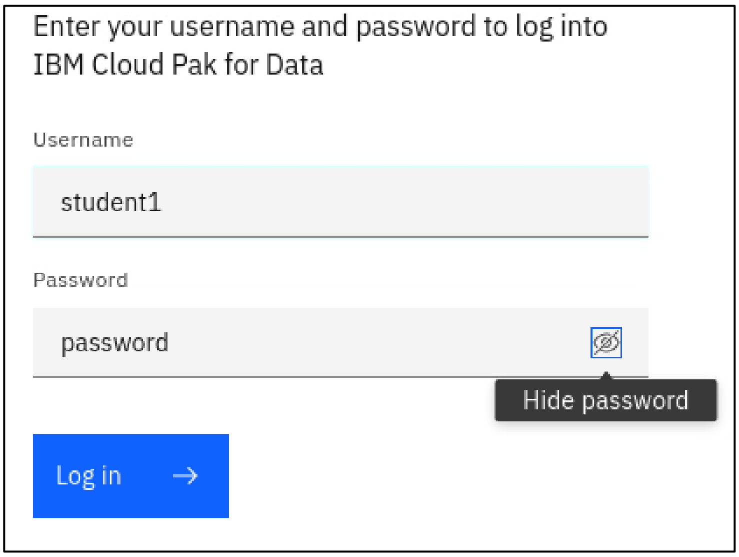
Enter your username and password to log into IBM Cloud Pak for Data

© IBM Corp 2020. All rights reserved. See product licensing for details.

You are now working in your virtual environment. Please take a moment to identify the different windows. In this environment, please take care when closing a browser window.

2. Enter **your student number** as username, and **password** as password.

   The results appear similar to the following (for student1, with the password shown):



Enter your username and password to log into IBM Cloud Pak for Data

Username

student1

Password

password

3. Click **Log in**.

2.2 Create an analytics project, import data to the project, and add a Modeler flow to the project.

An analytics project organizes assets such as data files, Modeler flows, and notebooks.

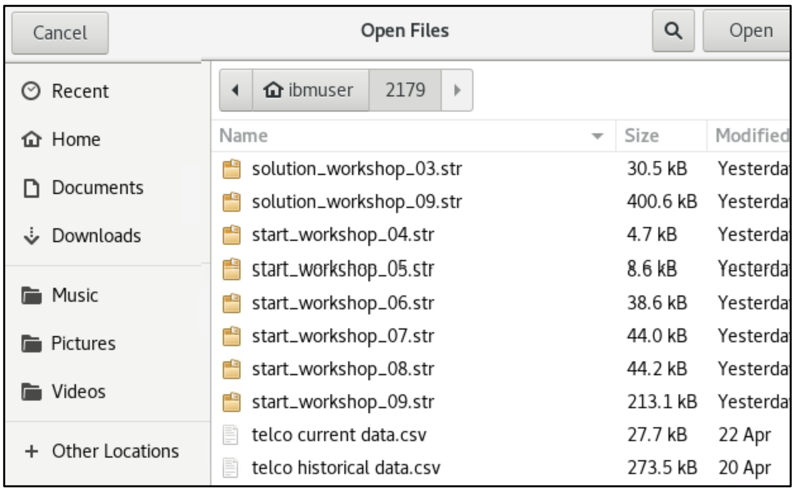
1. Click the **Navigation** ☰ menu at the top left, click **Projects** to expand if it is not already expanded, and then click **All projects**.
2. Click **New project**.
3. In the **Create a new project** window, validate that you will create an **Analytics project**, and then click **Next**.
4. Click **Create an empty project**.
5. In the **Name** box, type **predict churn project**, and then click the **Create** button at the bottom right.
6. Click the **Assets** tab to validate that you don't have any asset in this new project.

   You will add the datasets, used later in the workshops, to the project.

7. At the top right, click the **Find and add data** 01 00 icon to bring up the **Data** pane. (Note: This pane might already show.)
8. In the **Data** pane, click the **Load** tab, if not already selected.
9. Click **browse**.

10. Navigate to the **2179** folder, if that folder is not already open.

    The results appear similar to the following:



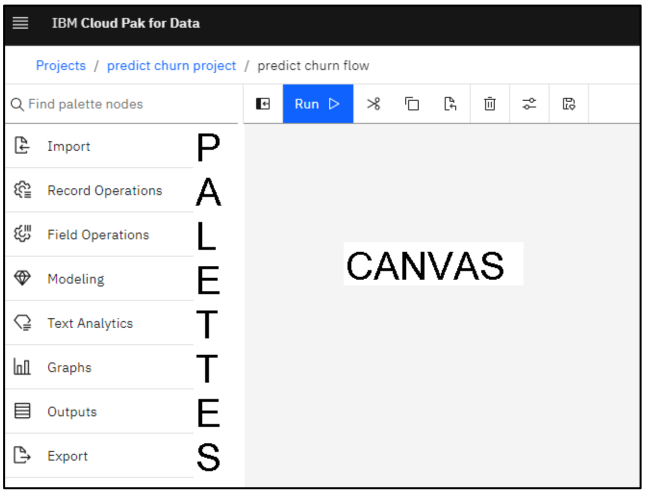
Name	Size	Modified
solution_workshop_03.str	30.5 kB	Yesterda
solution_workshop_09.str	400.6 kB	Yesterda
start_workshop_04.str	4.7 kB	Yesterda
start_workshop_05.str	8.6 kB	Yesterda
start_workshop_06.str	38.6 kB	Yesterda
start_workshop_07.str	44.0 kB	Yesterda
start_workshop_08.str	44.2 kB	Yesterda
start_workshop_09.str	213.1 kB	Yesterda
telco current data.csv	27.7 kB	22 Apr
telco historical data.csv	273.5 kB	20 Apr

11. Select **telco current data.csv**, press the **<Ctrl>** key, and then select **telco historical data.csv**.
12. Click **Open**.

    The data files are uploaded to the project and are listed under Data assets.

13. Click the **Find and add data** icon to hide the **Data** pane.
14. Click the **Add to project** button at the top, and then click **Modeler flow**.
15. In the **Name** box, type **predict churn flow**, and then click **Create**.
16. If, on the right hand side, a pane displays that reads **Drop files here or browse for files to upload**, hide it by clicking the **Data** icon at the top right.

    Once the Data pane is cleared, the results appear similar to the following (with the location of the "canvas" and the "palettes" illustrated; your canvas will read "Get started Drag a node ..:")



The largest area of the window is the canvas. This is where you will create your flow.

The palettes organize the data science workflow from importing data (the Import palette), preparing data for modeling (Record Operations and Field Operations palettes), building machine learning models (the Modeling palette), Text Analysis, producing graphical and tabular output (the Graphs and Outputs palettes), and exporting the data to other applications (the Export palette).

2.3 Explore the user interface (dry run).

In this task you will simulate importing data, preparing it for analyses, and producing output. It will be a dry run because no actual data will be used. The only objective is that you will get familiar with the interface.

First, assume that you want to import a comma-separated text file. The Import palette contains the nodes to import data.

1. Click the **Import** palette to expand it.

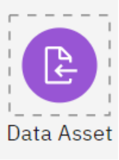
   The Data Asset node in the Sources palette is designed to import comma-separated text files. If you want to have more information about a node, you can point to it. You will get more information about the Data Asset node.

2. Point to the **Data Asset** node and take a moment to read the description.

   Double-clicking a node in a palette will add it to the canvas.

3. Double-click the **Data Asset** node.
4. Ensure that the **Data Asset** node has focus (it shows a dashed line around the node). If the **Data Asset n**ode does not have focus, click it to give it focus.

   The results should appear as follows:

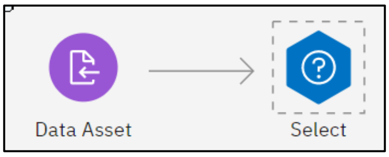
   

5. Click the **Import** palette to collapse it.

   As mentioned, this will be a dry run. So, just imagine that you have now imported data. Next, suppose that you want to select records, which is a record operation.

6. Click the **Record Operations** palette to expand it.
7. Double-click the **Select** node to add it to the flow.

   The results appear similar to:

   If you add a node from a palette to the canvas, it will automatically be added downstream from the node that has focus. In this example, the Select node is automatically added downstream from the Data Asset node.

   The flow indicates that data flows from the Data Asset node to the Select node: you start by importing data into Modeler and then you will select records.

   There is no rule about the lay-out of the nodes on the canvas, other than that your flow should be neat. So, in the remainder of this course, please feel free to lay-out the flow according to your own preferences (provided it is neat).

8. Click the **Record Operations** palette to collapse it.

   Next, assume that you want to derive a new field, which is a field operation.

9. Ensure that the **Select** node has focus (click it once to give it focus; the node then has a dashed line around it).
10. Click the **Field Operations** palette to expand it.
11. Double-click the **Derive** node to add it to the flow.

The flow indicates that you import data, then select records, and then derive a new field.

Normally, you would open a node and specify the settings. Let's see how you can open a node.

12. Double-click the **Derive** node that you have on the canvas to open it.

    A pane displays at the right side, where you can make your specifications, such as specifying the name of the field that you would like to derive.

13. Click **Cancel** to close the **Derive** pane.
14. Click the **Field Operations** palette to collapse it.

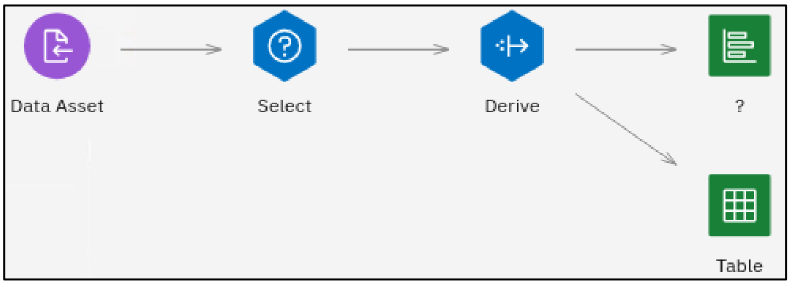
    Next, suppose that you want to chart the new field.

15. Ensure that the **Derive** node has focus.
16. Click the **Graphs** palette to expand it.
17. Double-click a **Distribution** node to add it to the flow.
18. Click the **Graphs** palette to collapse it.

    Assume that you also want to view the data, which is the Table node in the Outputs palette.

19. Click the **Outputs** palette to expand it.
20. Double-click a **Table** node to add it to the flow.

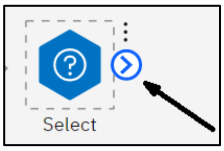
    The results appear similar to the following:



The Table node is not added downstream from the Distribution node but downstream from the Derive node because the data sits at the point of the Derive node (the Distribution only charts data).

So far, nodes were automatically connected when you added them to the canvas. It might happen that a new node is not automatically connected. To demonstrate how to connect nodes, you will first delete a node, say the Select node. You will then add it again.

21. Click the **Select** node to give it focus.
22. Press the <**Delete**> key.
23. On the canvas, click the **Data Asset** node to give it focus.
24. Click the **Record Operations** palette to expand it.
25. Double-click the **Select** node to add it downstream from the **Data Asset** node.
26. Point to the right side of the **Select** node, so that the **Connect** icon appears as indicated in the figure below.



27. Point the mouse to the **Connect** icon and then drag and drop the **Connect** icon onto the **Derive** node.

    The nodes are connected again.

Finally, you will delete all nodes so that you will start with an empty flow in the next task.

28. Press<**Ctrl-A**> to select all nodes on the canvas.
29. Press the <**Delete**> key to delete all nodes.
30. Leave Modeler open for the next workshop.

That completes this workshop. You have familiarized yourself with Modeler.

Note, that going forward:

- The instruction will just be to add a certain node from a certain palette (click the palette to expand it, and then double-click the node to add it to the flow)
- The instruction will just be to open a node (double-click the node which is already on the canvas)
- The instruction will just be to set focus on a node on the canvas (click it once)
- Point the cursor to a node in a palette if you want to know more about the node
- Please recall that numbered paragraphs provide instructions; non-numbered paragraphs provide context to the instructions

3 Build a machine learning model and apply it to current customers to identify who is likely to churn

In this workshop, you will see what Modeler is used for: to build models and apply them to new data. You will also learn that a number of requirements must be met to build *useful* models.

The business question is about a company named Amsel Telco. Amsel Telco is faced with many customers cancelling their subscription, which costs the company a significant amount of revenues. Therefore, the company wants to use machine learning models to predict who is likely to cancel and once such a model is in place, apply it to the current customers (who can then be targeted, for example, in a marketing campaign).

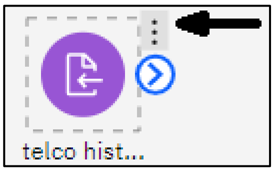
3.1 Import the data for modeling.

You have already uploaded the data to the project previously; you will now import it to Modeler.

1. From the **Import** palette, add the **Data Asset** node to the canvas (recall that you can double-click a node from a palette to add it to the canvas).
2. Open the **Data Asset** node (recall that you can double-click the node to open it).
3. Click **Change data asset**.
4. At the left side, expand **Data assets**, select **telco historical data.csv**, and then click **OK**.

   You will use the default settings to import the data.
5. Click **Save**, to return to the flow.

   You will validate the data import by previewing the data.
6. Hover the mouse over the top right part of the **telco hist**... node, so that three vertical dots appear, as indicated in the following figure.

7. Click the **three vertical dots**, and then click **Preview** from the drop-down menu.

   The preview displays the first 10 records of the dataset. It stores data of a telecommunications firm, containing demographics together with information about who cancelled their account, which is referred to as churn. It should be noted that the data is fictitious.
8. Scroll down and to the right, and observe that you have some empty cells.

   An empty cell represents missing information.
9. Scroll to the top, and then all the way to the right.

   The CHURN field flags whether the customer stayed (CHURN = No), or whether the customer has cancelled (CHURN = Yes). This is the field that you will predict shortly, based on demographical information and usage.

   Notice the RECEIVED_LETTER_OF_THANKS field. Those who have cancelled their subscription received a letter of thanks. The field is empty for those who are still with the company.
10. Click the **X** at the top right to close the **Data preview** window.

    To examine the entire dataset, you will use a Table node, which is located in the Outputs palette.
11. Ensure that the **telco hist ...** node has focus (click it if not).

12. From the **Outputs** palette, add a **Table** node to the flow.

    The Table node is added downstream from the node that imports the data.

13. Hover the mouse over the top right area of the **Table** node, click the **three vertical dots**, and then click **Run**.

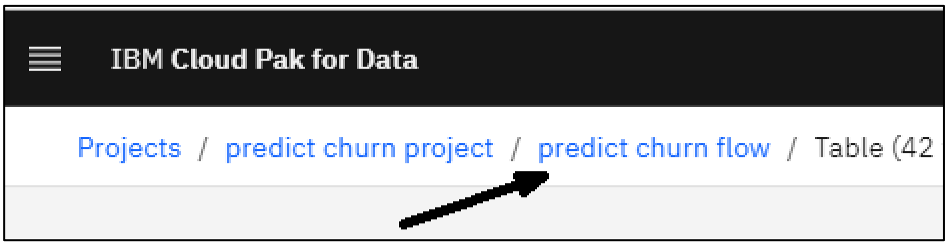
    The Table output item is listed in the Outputs tab at the right side. The description tells you that the dataset includes 42 fields and 1,000 records.

    Note: If the Outputs tab with the output item does not show, click the View outputs and versions ⟲ icon at the top right.

14. To the right of the **Table (42 fields, 1,000 re...)** output item, click the **Eye** 👁 icon to view the output.

    The Table output clearly shows that you have 42 fields and 1,000 records and if you want, you can scroll through the data to get familiar with it. You will not do that here.

15. Click **Return to flow** at the top left to return to the flow. Alternatively, click **predict churn flow** in the **bread crumbs**, as illustrated in the following figure.

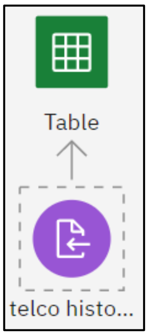


IBM Cloud Pak for Data

Projects / predict churn project / predict churn flow / Table (42

16. Reposition the nodes, so that the **Table** node is above the **telco histo...** node (you can drag a node to reposition it).

    The results appear similar to the following:

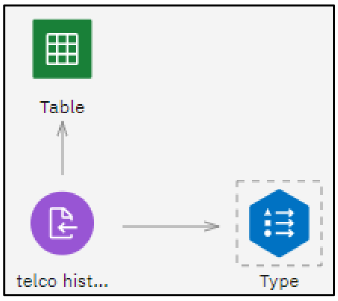


3.2 Set the roles for modeling.

Having imported the data, the next step is to build a model to predict churn. Predictors have to be selected and the target field (the field that you want to predict) needs to be specified. This is done in a Type node, located in the Field Operations palette.

1. Ensure that the **telco hist...** node has focus.
2. From the **Field Operations** palette, add a **Type** node to the flow.

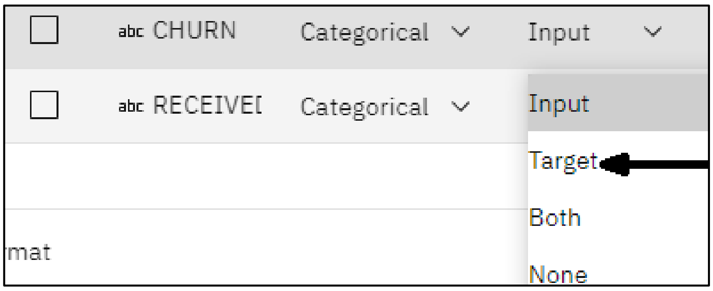
   The results appear similar to the following:



3. Open the **Type** node.

   You will review the Type settings later. The only thing that you will set now, is the target, the field that must be predicted, which is CHURN. In data science, the target is also referred to as label.
4. Scroll to the **CHURN** row (the last-but-one field; instead of scrolling, you can also press the **<Arrow down>** key; or, sort the fields alphabetically by clicking the **Field** column header to locate the CHURN field easier).

   You will designate CHURN as the target (the field to predict) in the Role column.
5. Click the cell in the **CHURN** row, **Role** column where it currently reads **Input**, and then click **Target** from the drop-down menu, as shown in the following figure.



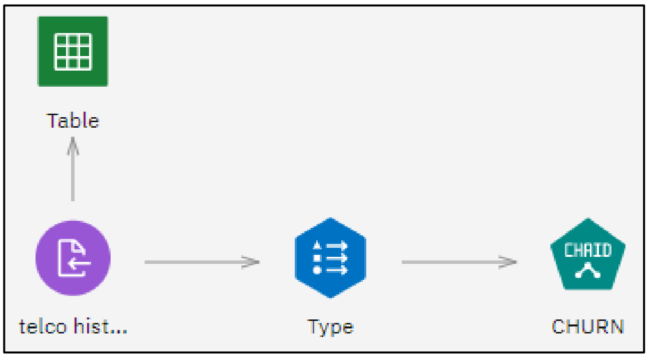
   The fields that will be used to predict the target have role Input. Here, all other fields have role Input so they will all be used to predict CHURN.
6. Click **Save**, to return to the flow.

3.3 Build a machine learning model.

Now that you have specified predictors and target, you will build a machine learning model. CHAID, a commonly used algorithm in Modeler, will be demonstrated here.

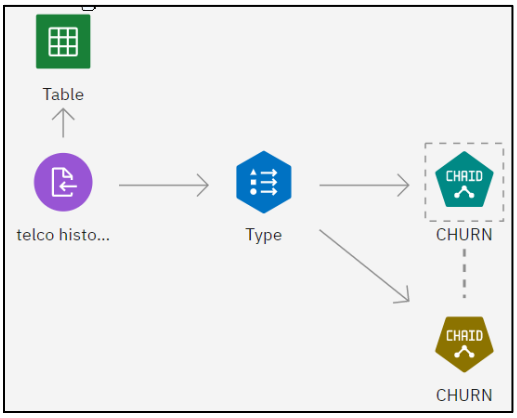
1. Ensure that the **Type** node has focus.
2. Expand the **Modeling** palette.
3. Locate the **CHAID** node, and then add it to the flow.

   The results appear similar to the following:

   The CHAID node automatically recognizes the target field, CHURN. (If not, open the Type node and set the role of CHURN to Target.)
4. Click the **three vertical dots** at the top right of the **CHAID** node, and then click **Run**.
5. If a **Messages** window appears, close it. (Note: in the remainder of this course, if the Messages window appears, please close it.)

   The results appear similar to the following:

   A new node is added to the canvas, also labeled CHURN. This is the model nugget; it stores the rules found by CHAID. More in general, a model nugget stores the model (the outcome of running a machine learning algorithm).

   When data flows through the model nugget, the rules contained in the model nugget are applied to each record and the predictions made by the model will be added to the data. Previewing the data will show these new fields.
6. Preview the data at the newly added **model nugget** (click the t**hree vertical dots** on the model nugget, and then click **Preview**).
7. Scroll all the way to the right in the **Data preview** window.

   The predicted value is stored in $R-CHURN; $RC-CHURN stores the confidence for the prediction. For example, you can be 0.996, or 99.6%, confident that the first customer will cancel. Looking at the CHURN field, this is a correct prediction.

8. Click the **X** at the top right to close the **Data preview** window.

3.4 Assess the accuracy of the model.

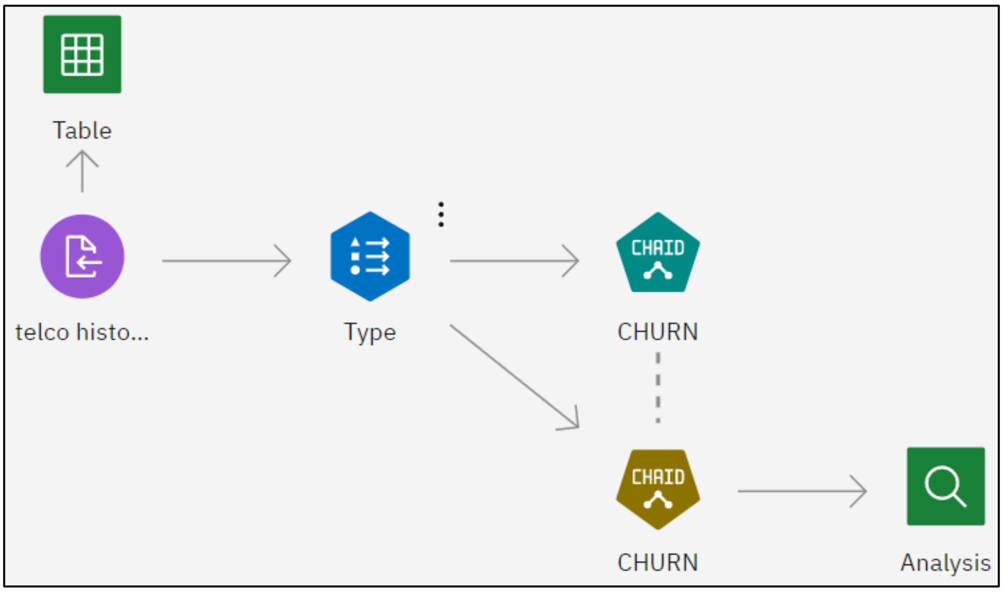
At this point, you have both the actual CHURN and the predicted churn ($R-CHURN). So, you can see if the actual and predicted value match and compute the percentage of correct predictions.

The percentage of correct predictions is referred to as accuracy.

The Analysis node lets you assess the accuracy of the model.

1. Ensure that the **model nugget** has focus.
2. From the **Outputs** palette, add an **Analysis** node downstream from the **model nugget.**

   The results appear similar to the following:



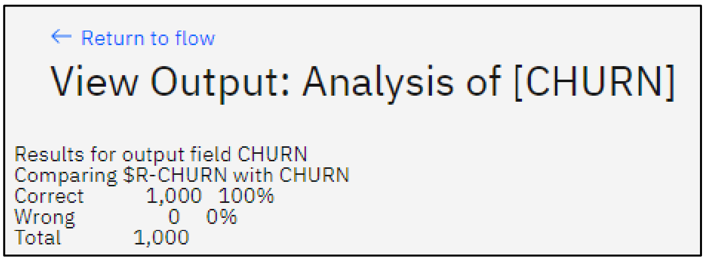
3. Run the **Analysis** node (click the **three vertical dots** at the top right of the node, and then click **Run**).

   An Analysis output item is added to the Outputs tab.

   Note: if the Outputs tab with the output items does not show, click the View outputs and versions ⟲ icon.

4. To the right of the **Analysis of [CHURN]** output item, click the **Eye** 👁 icon to view the output.

   The results appear similar to the following:



← Return to flow

View Output: Analysis of [CHURN]

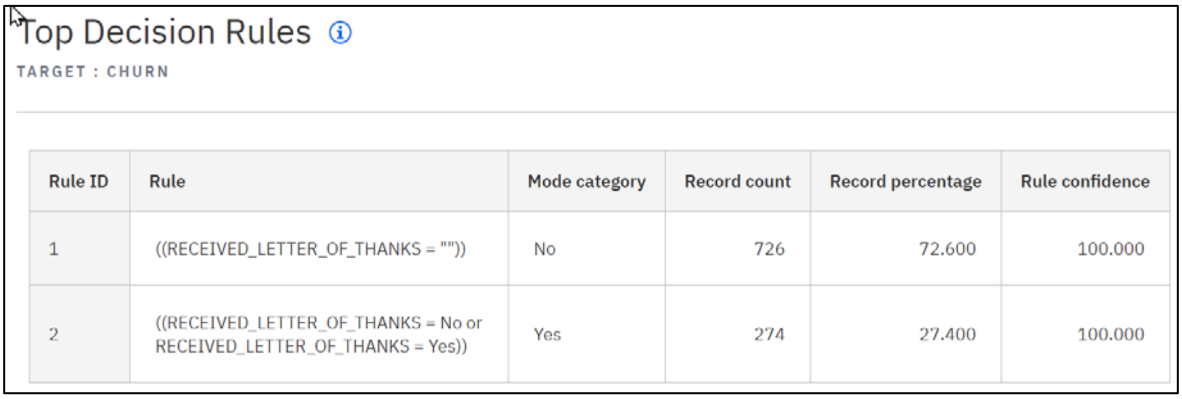
```
Results for output field CHURN
Comparing $R-CHURN with CHURN
Correct        1,000  100%
Wrong              0    0%
Total          1,000
```

The accuracy of the CHAID model is 100%. So, the predicted churn values ($R-CHURN) match the actual churn values (CHURN) perfectly.

A 100% accuracy is usually a signal that something went wrong in model building. Let's examine the model.

5. Click **Return to the flow**.
6. Hover the mouse over the top-right area of the **model nugget**, click the **three vertical dots**, and then click **View Model**.
7. At the left side, click **Top Decision Rules**.

   The results appear similar to the following:



Top Decision Rules

TARGET : CHURN

Rule ID	Rule	Mode category	Record count	Record percentage	Rule confidence
1	((RECEIVED_LETTER_OF_THANKS = ""))	No	726	72.600	100.000
2	((RECEIVED_LETTER_OF_THANKS = No or RECEIVED_LETTER_OF_THANKS = Yes))	Yes	274	27.400	100.000

The Rule column states the condition, and the Mode category column tells you what the predicted value of the target is for the customers who meet the condition.

The CHAID algorithm has found two decision rules. The first decision rule states that customers who have an empty value for RECEIVED_LETTER_OF_THANKS, are predicted to not cancel their subscription.

The second rule states that customers who did receive a letter of thanks are predicted to cancel their subscription.

Now, receiving a letter of thanks is a *consequence* of having cancelled, not a cause. In time, receiving a letter of thanks will occur after having cancelled. In short, this is a useless model.

In this example, it is quite trivial that such a field should be excluded from model building. But in a dataset with, for example, 481 fields originating from different databases it might be less obvious which fields are recorded prior to the event of cancelling the subscription and which fields are a consequence of cancelling. In other words, you need *domain knowledge* to choose the relevant predictors.

You will redo the modeling, this time excluding RECEIVED_LETTER_OF_THANKS from the analysis. Recall that you can set the modeling roles in a Type node.

8. Return to the flow by clicking **predict churn flow** in the **bread crumbs**.
9. Open the **Type** node.
10. Scroll to the **RECEIVED_LETTER_OF_THANKS** row (the last field; instead of scrolling, you can also use the **<Arrow down>** key; or, sort the fields alphabetically by clicking the **Field** column header to locate the field easier).
11. In the **RECEIVED_LETTER_OF_THANKS** row, click the cell in the **Role** column where it currently reads **Input**, and then click **None**.

    RECEIVED_LETTER_OF_THANKS is no longer used to predict CHURN.

12. Click **Save**, to return to the flow.

    You need to rerun the CHAID algorithm to build a model that will exclude RECEIVED_LETTER_OF_THANKS as a predictor.

13. Run the **CHAID** node labeled **CHURN** (the upper node, not the CHAID model nugget). (If a Messages windows displays after running the node, close it.)

    Because the CHAID algorithm and the CHAID model nugget are linked (the dashed line between the two), running the CHAID algorithm will update the CHAID model nugget automatically.

    You will validate that the decision rules have changed.

14. Hover the mouse over the top-right area of the **model nugget**, click the **three vertical dots**, and then click **View Model**.

15. At the left side, click **Top Decision Rules**.

    The decision rules are now non-trivial.

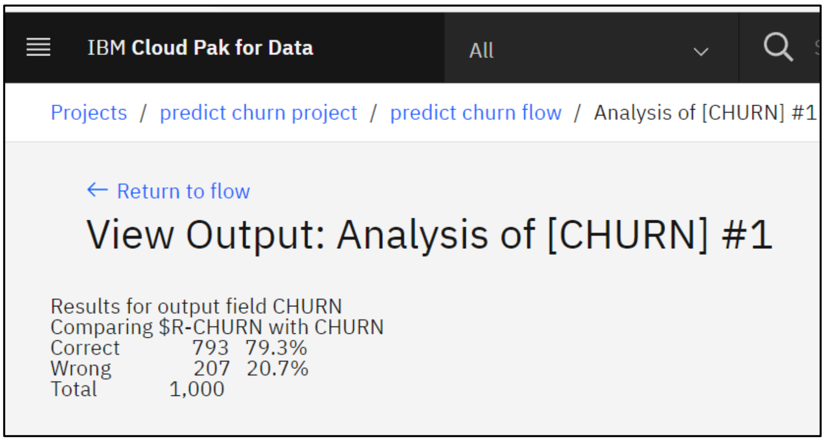
16. Return to the flow by clicking **predict churn flow** in the **bread crumbs**.

    You will run the Analysis node to see how accurate the updated model is.

17. Hover the mouse over the top-right area of the **Analysis** node, click the **three vertical dots**, and then click **Run**.

18. To the right of the **Analysis of [CHURN] #1** output item, click the **Eye** 👁 icon to view the output.

    The results appear similar to the following:



IBM Cloud Pak for Data
All
Projects / predict churn project / predict churn flow / Analysis of [CHURN] #1
← Return to flow

View Output: Analysis of [CHURN] #1

```
Results for output field CHURN
Comparing $R-CHURN with CHURN
Correct        793  79.3%
Wrong          207  20.7%
Total        1,000
```

    The new model is about 80% accurate.

19. Return to the flow.

3.5 Apply the model to predict the behavior of the current customers.

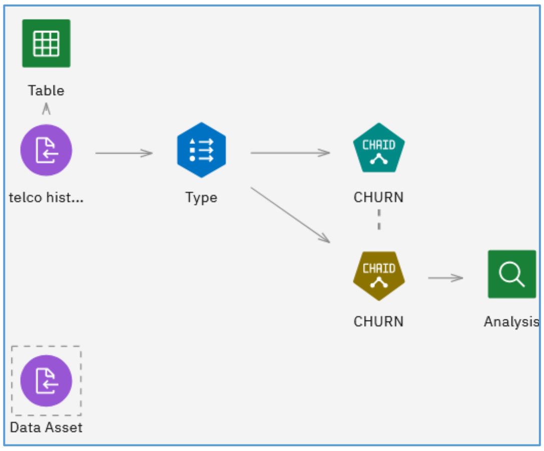
In the previous tasks, you have built a model on historical data to find patterns in who do and who do not cancel their subscription. These rules are stored in a model, represented by the model nugget in Modeler.

Having a model (in the form of the model nugget) in place, it can be used to predict the behavior of the current customers. In data science, this is referred to as *scoring data*.

In this example, data for the current customers is stored in another comma separated text file, telco current data.csv. You have already uploaded that file to the project in the first task, so you will import it now to Modeler.

1. From the **Import** palette, add a **Data Asset** node to the canvas, and, to have a neat flow, position it below the first Data Asset node.

   The results appear similar to the following:



2. Open the **Data Asset** node.
3. Click **Change data asset**.
4. At the left side, expand **Data assets**, select **telco current data.csv**, and then click **OK**.
5. Click **Save**, to return to the flow.

   You will validate the data import by previewing the data.

6. Preview the data, and then scroll all the way to the right in the **Data preview** window.

   This data includes the same fields as the historical data. In this example, CHURN and RECEIVED_LETTER_OF THANKS are also included in the data, and all values are empty. In practice, these fields might not even be included in the data.

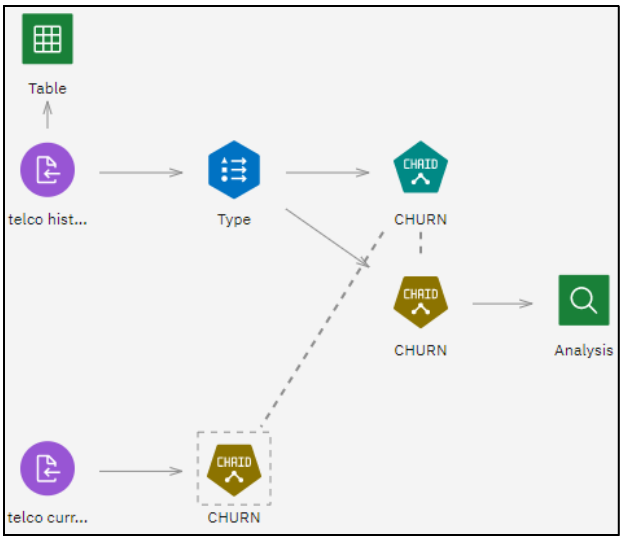
7. Click the **X** at the top right to close the **Data preview** window.

   To add the predictions from the model to the data, all what is needed is the model nugget because that stores the model (the decision rules in this case). When data flows through the model nugget, the rules that are contained in the model nugget will be applied to the data.

   You will copy the model nugget that you already have on the canvas.

8. Click the **CHAID model nugget** that is already on the canvas, to give it focus.
9. Click the **Copy** icon at the top.
10. Click the **Paste** icon at the top.

11. Connect the **telco curr ...** node to the **model nugget** named **CHURN**, that you just pasted onto the canvas, and reposition the node somewhat, similar to the following:



The data of the current customers will be imported and the model will be applied to this data.

Let's examine the scored data.

12. Preview the data at the **model nugget** that you just pasted and added downstream from the **telco curr...** node.
13. In the **Data preview** window, scroll all the way to the right.

    Like before, two fields are added to the data, the prediction ($R-CHURN), and the confidence for that prediction ($RC-CHURN). For example, you can be about 77% confident that the first customer will cancel.

    At this point, you could export the data of those who are predicted to leave, to target them in a marketing campaign later. You will skip that for now.

14. Click the **X** at the top right to close the **Data preview** window.

    You have built a model and applied it to new data. This went quite smoothly.

    However, there are more things to consider than we have done so far. Let's open the Type node.

15. Open the **Type** node.

    CUSTOMER_ID has role Input, so is used as a predictor. However, it is a record identifier only and should be excluded from the analysis. (Please do *not* take action on this right now; you will do that in a next workshop.)

16. Scroll to **POSTAL_CODE**.

    POSTAL_CODE is typed as a "Continuous" field, while its values just represent different postal code areas. Because it is continuous, the CHAID algorithm (and actually all algorithms) will compute the mean POSTAL_CODE behind the scenes, which makes no sense. So, the model that you have built now does not make sense. (Please do *not* take action on this right now; you will do that in a next workshop.)

    Also, you might try to improve the accuracy if you derive new predictors. Or, you could try another model than CHAID to improve the accuracy. (Please do *not* take action on this right now; you will do that in a next workshop.)

    Furthermore, in model building, one should not build the model and evaluate the model on the same dataset, as we have done here. (Please do *not* take action on this right now; you will do that in a next workshop.)

17. Click **Cancel** to close the **Type** node.

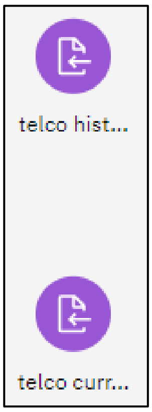
This workshop sets the stage for the next workshops, where you will basically redo the analysis, but with more attention to the details so that the model will make sense. You will:

- Examine the type of the fields (Workshop 4)
- Replace out-of-range values and missing values (Workshop 5)
- Select the most important predictors (Workshop 6)
- Split the data into a dataset to build models and a dataset to evaluate models (Workshop 7)
- Run more models and identify the best model (Workshop 8)
- Apply the best model to score the current customers (Workshop 9)

You will prepare for the next workshop.

18. Delete all nodes, except for the two **Data Asset** nodes (**telco histo**... and **telco curr...**). (Pressing the **<Ctrl>** key lets you select multiple nodes; then press the **<Delete>** key.)

    The results appear similar to the following:



19. Leave Modeler open for the next workshop.

That completes this workshop. You have built a machine learning model to predict customer churn, and have applied that model to the current customers. It appeared that building models is straightforward, but domain knowledge is required. Based on domain knowledge, you need to select relevant predictors, among others.

Note, that going forward:

- The instruction will just be to run a node (hover the mouse over the top right area of the node, and then click Run)
- The instruction will just be to return to the flow (click Return to the flow if that option is available, or click the flow in the bread crumbs)
- The instruction will just be to open an output item (in the Outputs tab, click the Eye icon to the right of the output item)
- The instruction will just be to close the Data preview window (click the X at the top right to close the Data preview window)
- Please recall that numbered paragraphs provide instructions throughout the course; non-numbered paragraphs provide context to the instructions

4 Set measurement levels

One of the issues that were mentioned in the previous workshop was that fields like CUSTOMER_ID, POSTAL_CODE, and REGION were treated as quantitative information because they are numeric.

In this workshop, you will set the correct "measurement level" of these fields

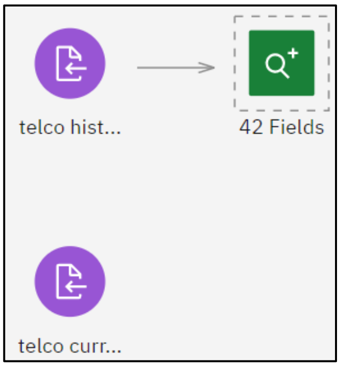
4.1 Audit the data.

In the previous workshop, you have imported a data file into Modeler. It stores data of a telecommunications firm, containing demographics together with information about who cancelled their subscription, which is referred to as churn. It should be noted that the data is fictitious.

First, let's examine the data. A useful node to examine the data is the Data Audit node.

1. Ensure that the **telco hist...** node has focus.
2. From the **Outputs** palette, add a **Data Audit** node to the flow.

   The results appear similar to the following:

3. Run the **Data Audit** node.
4. Open the **Data Audit** output item.

   Because the storage of CUSTOMER_ID is numeric, it is typed as a continuous field. Therefore, Modeler treats it as quantitative information and will report on the mean of this field. However, CUSTOMER_ID is an identifier only and has to be excluded from all analyses. (Please do *not* take action on this right now; you will do that in the next task.)

5. Scroll down so you can view **POSTAL_CODE** and **REGION**.

   Both fields store categorical information (for example, REGION 1 represents "North") but because they are numeric, they are typed as continuous and their means are computed. (Please do *not* take action on this right now; you will do that in the next task.)

6. Return to the flow.

4.2 Set measurement levels.

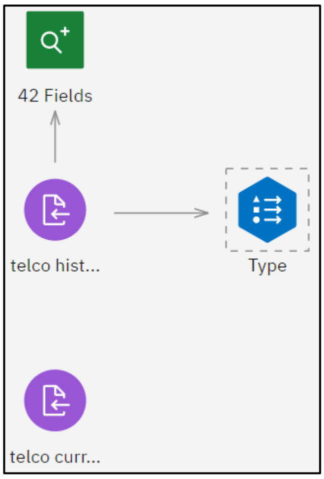
Modeler uses the measurement level of a field to determine which statistics (or which analyses more in general) are meaningful.

CUSTOMER_ID is a record identifier only; in Modeler, the "Typeless" measurement level is used for fields that you do not want to include in any analysis. So, you will set the measurement level of CUSTOMER_ID to Typeless.

Measurement levels are modified in a Type node.

1. Reposition the **Data Audit** node, so that it is above the **telco hist...** node.
2. Ensure that the **telco hist...** node has focus.
3. From the **Field Operations** palette, add a **Type** node downstream from the **telco hist....** node.

   The results appear similar to the following:



4. Open the **Type** node.
5. In the **CUSTOMER_ID** row, click the cell in the **Measure** column where it currently reads **Continuous**, and then click **Typeless** from the drop-down menu.

   When you set the measurement level of a field to typeless, its role is automatically set to None. Typeless fields play no role in modeling or in any other analysis.

   Currently, POSTAL_CODE and REGION are typed as continuous, while regions and postal codes are only codes to represent different geographical areas. So, they should be typed as categorical.
6. In the **POSTAL_CODE** row, click the cell in the **Measure** column where it currently reads **Continuous**, and then click **Categorical**.
7. In the **REGION** row, click the cell in the **Measure** column where it currently reads **Continuous**, and then click **Categorical**.

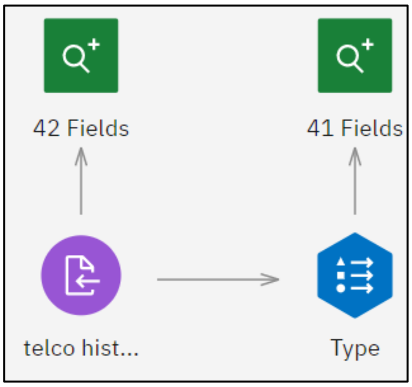
   After setting the correct measurement levels, you will read the data.
8. Click the **Read Values** button.

   Examining the results for REGION, four values are listed in the Values column, rather than the range of values (minimum and maximum) as for continuous fields.
9. Click **Save**, to return to the flow.

You will check the results by running another Data Audit node.

10. Ensure that the **Type** node has focus.
11. From the **Outputs** palette, add a **Data Audit** node downstream from the **Type** node, and then reposition the nodes somewhat.

    The results appear similar to the following:

    Typeless fields will not be audited, which explains the difference of 1.

    Note: From this point going forward, the second data file (telco curr...) will not be shown in screenshots, until it is relevant.

12. Run the **Data Audit** node.
13. Open the **Data Audit** output item.

    You can validate that CUSTOMER_ID is not included in the report; POSTAL_CODE and REGION are now handled as categorical fields (for example, the mean is not computed for them).

14. Leave the **Data Audit** output window open for the next workshop (you will examine it further in the next workshop).

That completes this workshop. You have set the measurement level of fields. Modeler uses this information to choose the right analysis or machine learning model. In other words: when your fields have the correct measurement level, you will be able to use all machine learning models without running the risk of computing the mean on categorical fields behind the scenes. The concept of "measurement level" is the most important concept to ensure that your analysis is meaningful (does not produce nonsense results like computing mean postal code).

5 Automatically prepare the data for modeling

You will examine the Data Audit output more closely and observe that some fields have outlier values and missing data. You will then use the Auto Data Prep node to replace those outliers and missing data.

5.1 Use the Auto Data Prep node.

At this point, you should still have the Data Audit output open.

1. In the **Data Audit** output, examine **HOUSEHOLD_SIZE** (the fifth field in the report).

   Note that the maximum household size is 81. Examining the chart at the left side, the tiny bar at the far right represents this outlier value (note: this might be hard to see).

   The last column in the Data Audit output reports on the number of valid values for each field. For example, HOUSEHOLD_SIZE has 996 valid values, in a dataset of 1,000 records. So, four records don't have a valid value for HOUSEHOLD_SIZE.

   Other fields also have outlier values and/or missing data.

2. Return to the flow.

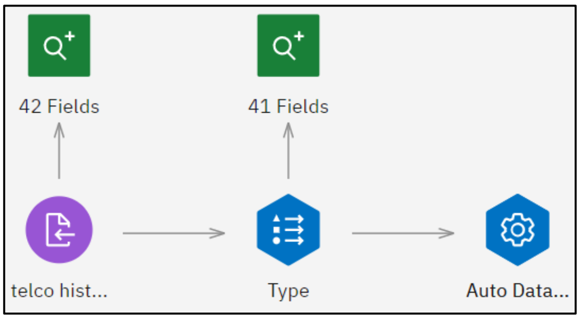
   There are several ways to deal with outlier values and missing data.

   The Auto Data Prep node is useful to scan the data for missing values and outlier values, replacing them if required.

   The Auto Data Prep node can perform many more checks and transformations, but you will not use them in this workshop.

3. Ensure that the **Type** node has focus.
4. From the **Field Operations** palette, add an **Auto Data Prep** node to the flow.

   The results appear similar to the following:



5. Open the **Auto Data ...** node.
6. Collapse the **Clear Analysis** tab.
7. Expand the **Objectives** tab.

   You will customize the analysis.

8. Click **Custom Analysis**, and then collapse the **Objectives** tab.

   By default, fields that store dates or times will be transformed so they can be included in modeling later. You will not use this option.

9. Expand the **Prepare Dates & Times** tab, disable the **Prepare dates and times for modeling** option, and then collapse the **Prepare Dates & Times** tab.
10. Expand the **Exclude Input Fields** tab. ("Inputs" refers to the fields used to predict a target.)

    You can automatically exclude fields with too many missing values (like RECEIVED_LETTER_OF_THANKS), categorical fields with too many categories (like POSTAL_CODE) or too many values in a single category. You will not use these options.

11. Disable the **Exclude low quality inputs fields** option.

12. Collapse the **Exclude Input Fields** tab, and then expand the **Prepare Inputs & Target** tab.

    You will focus on replacing outliers and missing values.

13. Under **Adjust the type of numeric fields (ordinal and continuous)**, disable the **Inputs** and **Target** option (you do not want to change the fields' measurement levels automatically).
14. Scroll down to **Reorder nominal fields to have the smallest category first, largest last**, and then disable the **Inputs** option (the order of categories of categorical fields should be left as is).
15. Scroll down to **Replace outlier values in continuous fields (recommended for input fields if they will be put on a common scale)**.
16. Enable the **Inputs** option.

    Outlier values in continuous fields will now be replaced (by a cutoff value of mean + 3 * standard deviation).

    Missing value replacement for input fields (the predictors) is used when the number of records available for modeling drops drastically because of the presence of missing values. Although that is not the case here, you will replace missing values.

17. Scroll down to **Continuous fields: replace missing values with mean**.

    By default, missing values in continuous fields will be replaced by the mean, which is what we want.

18. Scroll down to **Nominal fields: replace missing values with mode**.

    Missing values in "nominal" fields (a sub class of categorical fields) will be replaced by the mode, which is the most frequent value, which is what we want.

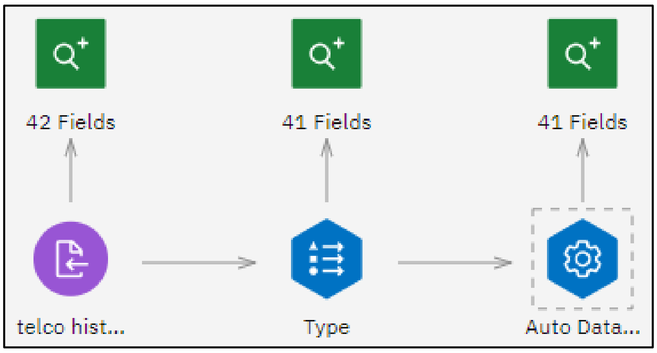
    Note that the measurement level of a field the method of replacement of a missing value (mean for continuous fields; mode for nominal fields), again emphasizing the importance of measurement levels.

19. Scroll down to **Put all continuous input fields on a common scale (highly recommended if feature construction will be performed)**, and disable this option.
20. Click **Save**, to return to the flow.

    You will run another Data Audit node to see the changes.

21. Ensure that the **Auto Data...** node has focus.
22. From the **Outputs** palette, add a **Data Audit** node to the flow, and reposition the node somewhat.

    The results appear similar to the following:



23. Run the **Data Audit** node.
24. Open the **Data Audit** output.
25. Scroll down to **HOUSEHOLD_SIZE transformed** (number 37 in the report).

    HOUSEHOLD_SIZE is changed in two ways: outlier values and missing values are replaced.

Outliers were replaced by the value of 5.924 (mean + 3 * standard deviation); the maximum value is now no longer an integer.

Missing values are replaced by the mean (because HOUSEHOLD_SIZE is continuous), and the number of valid records is now 1,000.

26. Return to the flow.
27. Leave Modeler open for the next workshop.

That completes this workshop. You have replaced outliers and missing values. How outliers or missing values in a field are replaced, depends on the measurement level of the field in question.

Hopefully, auto data preparation lets you build more accurate machine learning models later. There are many more options in the Auto Data Prep node that you could try out, but they are beyond the objectives of this course.

It should be noted that you can also create fields based on domain knowledge, or try mathematical transformations such as taking the square root of a field's values. Modeler provides much more functionality than we use in this course.

In data science, you can explore many different paths which makes that commonly 80% of all the work is about "wrangling" the data (preparing data for modeling).

6 Select the most important predictors ("features")

If you have many predictors (referred to as features here), model building can take a significant amount of time. Therefore, you might select the most important features before building models.

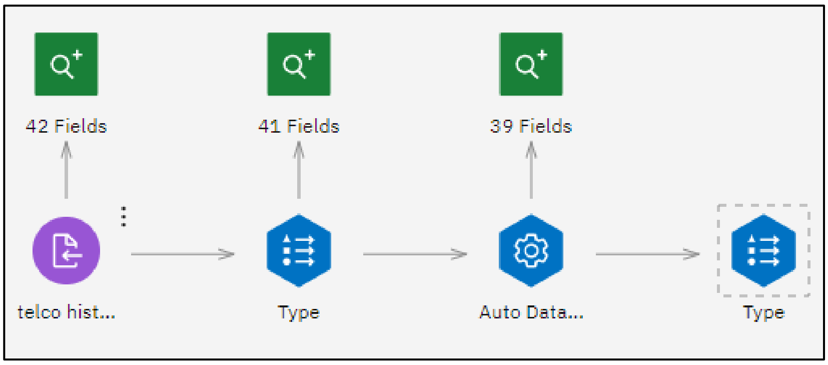
The Feature Selection node will select the fields which are most strongly associated with the target field, using the statistical criterion of "significance".

6.1 Select the most important predictors with the Feature Selection node.

At this point, you did not yet declare CHURN as target, so you will do that first. As before, the Type node lets you set the roles.

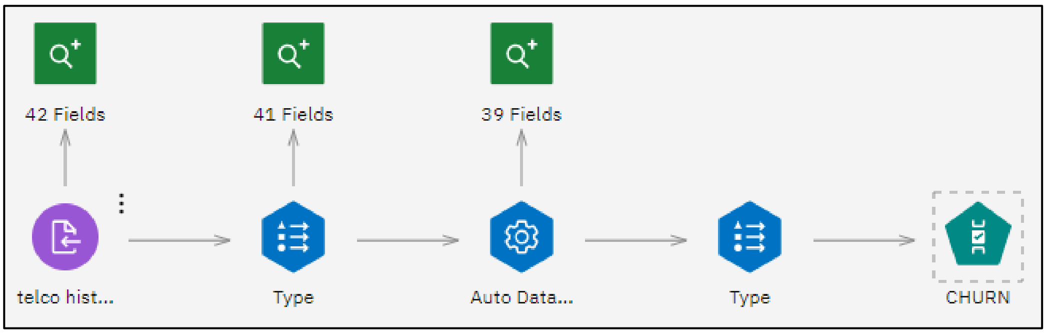
1. Ensure that the **Auto Data Prep** node is selected.
2. From the **Field Operations** palette, add a **Type** node downstream from the **Auto Data...** node.

   The results appear similar to the following:



3. Open the **Type** node.
4. Click the **Field** column header and sort the fields alphabetically, to better locate **CHURN**.
5. In the **CHURN** row, click the cell in the **Role** column where it currently reads **Input**, and then click **Target** from the drop-down menu.
6. Click **Save**, to return to the flow.
7. Ensure that the **Type** node has focus.
8. Expand the **Modeling** palette.
9. Locate the **Feature Selection** node, and then add it to the flow.

   The results appear similar to the following:



   The Feature Selection node recognizes CHURN as the target field.

10. Open the **Feature Selection** node.
11. Click the **Fields** tab to collapse it.

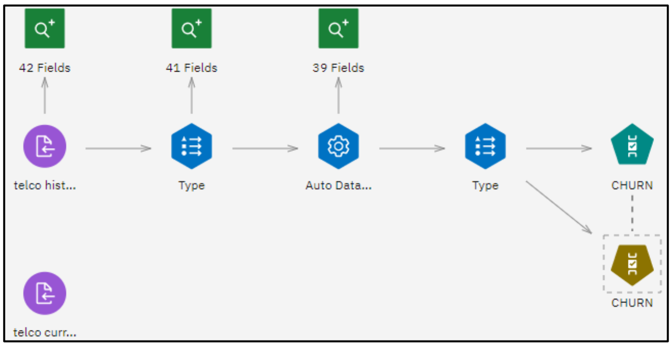
12. Click the **Options** tab to expand it (take care to expand the Options tab, not the Build Options tab).

    Fields are selected for their bivariate association with the target and, by default, a field with an importance value greater than 0.95 will be labeled important.

    You can change the cutoff values for the various categories. In very large datasets you might increase the cutoff values, otherwise most fields will be labeled as Important (statistical significance goes hand in hand with sample size). The current dataset of 1,000 records is not that large, so you will keep the default.

13. Click **Cancel**, to return to the flow.
14. Run the **Feature Selection** node.

    The results appear similar to the following:



A model nugget is generated that stores the results of running the Feature Selection node.

15. Double-click the **Feature Selection model nugget** to open it.

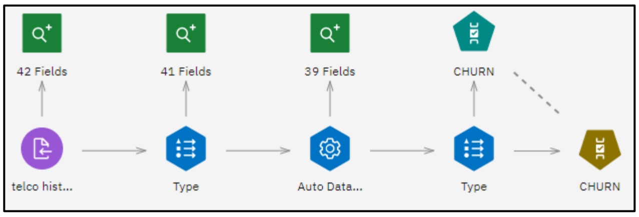
    The checked fields are flagged as important for the target (CHURN) and only these fields will be passed downstream from the Feature Selection model nugget.

    You can validate that 23 fields have been marked as important.

    The Feature Selection node will perform other checks as well. Four fields are excluded because they have too many records in a single category (for example, RETIRED); one field is excluded because it has too many missing values (RECEIVED_LETTER_OF_THANKS).

16. Click **Cancel**, to return to the flow.
17. Reposition the nodes, so that the flow of data is clearer.

    The results appear similar to the following:



18. Leave Modeler open for the next workshop.

That completes this workshop. You have selected the most important fields to predict CHURN, which will speed up building model later.

It should be mentioned that the Feature Selection node only assesses the relationship of each field separately with the target. It might happen that, say, a field AGE CATEGORY is not related to the target, and neither is a field GENDER. However, it might well be that young men cancel with much higher rates than others. Such "interaction effects" between the predictors will not be detected by the Feature Selection node. That said, if you have many predictors, Feature Selection can prove useful to reduce the number of predictors.

7 Partition the data into a training and testing set

A model should be built and evaluated on different datasets, the training and testing set respectively.

The records should be assigned randomly to the training set and testing set.

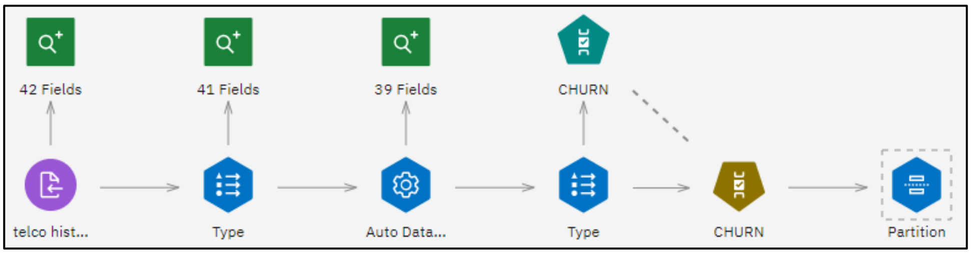
7.1 Use the Partition node.

The Partition node provides great flexibility in creating the training and testing set.

The Partition node will create a new field that flags whether a record belongs to the training set or to the testing set. Therefore, you will find the Partition node in the Field Operations palette.

1. Ensure that the **Feature Selection model nugget** has focus.
2. From the **Field Operations** palette, add a **Partition** node to the flow.

   The results appear similar to the following:



3. Open the **Partition** node.

   By default, 50% of the records is selected to train the model, and the other 50% to evaluate the model on unseen data, the testing set.

   In data science, more data is used to develop the model than is used to test the model. With moderate sized datasets, a typical split is between 70% and 80% for training and 20% and 30% for testing.

   You will use75 a training set of 75%.
4. In the **Training Partition(%)** text box, type **75**.
5. In the **Testing Partition(%) t**ext box, type **25**.
6. Click **Save**, to return to the flow.

   You will preview the data to validate that a new field is added to the data.
7. Preview the data at the **Partition** node.
8. Scroll all the way to the right in the **Data preview** window.

   A new field named Partition has been added to the data; it flags whether a records belongs to the training set or testing set.

   Nine out of the first ten records were assigned to the training set.
9. Close the **Data preview** window.
10. Leave Modeler open for the next workshop.

That completes this workshop. You have partitioned the data by adding a field that indicates whether a record is used to train the model or to test the model.

Note: Ideally, the testing set should not be involved in any data preparation, to simulate the situation that you will apply a model to new data eventually. Is that requirement met here? No, it is not because now the testing records are included in the auto data preparation step and in the feature selection step. The issue is, that the Auto Data Prep does not honor data partitioning (the Feature Selection node does). We can set up the flow differently, but that would involve somewhat more work. For convenience, we did not do that. But please take note that in practice, if you want to use the Auto Data Prep, the flow should be tweaked compared to what we have now.

8 Use automated modeling to predict CHURN

In this stage, you want to try out many models and see which of them performs best in terms of a certain evaluation metric such as accuracy.

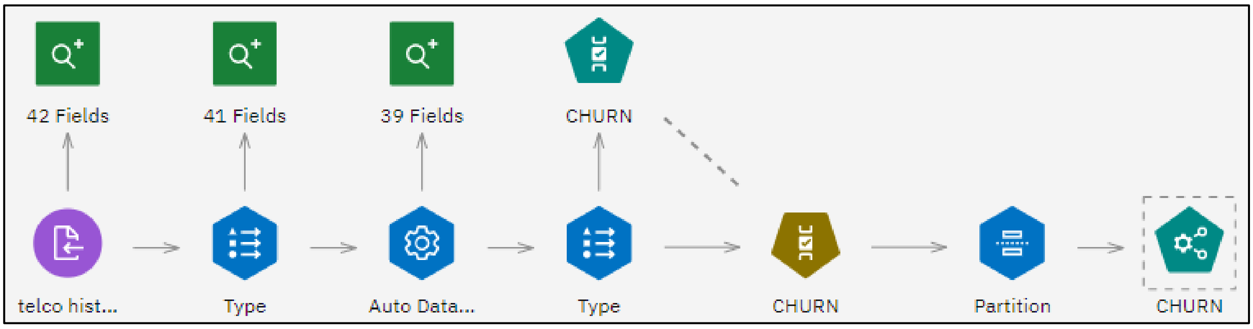
Because we have included a Partition node in the flow, the training data will be used to build the model and we can assess the accuracy of the model on the testing data.

8.1 Build models to predict a categorical target.

The Auto Classifier node lets you run multiple algorithms in a single run and identify the best model. (The Auto Classifier node is designed to predict a categorical target, such as CHURN in this example. If the target would be continuous, the Auto Numeric node would be the correct choice.)

1. Ensure that the **Partition** has focus.
2. From the **Modeling** palette, add the **Auto Classifier** node to the flow.

   The results appear similar to the following:



   To save time, you will run a selection of models.

3. Open the **Auto Classifier** node.
4. If the **Fields** tab is expanded, click it to collapse.
5. Click the **Build Options** to expand it.

   By default, if there is a partition, it will be honored, so models will be built on the training set and evaluated on the testing set.

6. Click the **Build Options** to collapse it, and then click the **Expert** tab to expand it.

   Only models relevant to predict a categorical target are listed. The more models selected, the longer the execution time will be. You will make a selection of models.

7. Disable the following model types: **C5**, **Decision List**, **Bayesian Network**, **Discriminant**, **LSVM**, **Tree-AS**, **XGBoost Linear**, **Quest**, and **Neural Net** (so, selected are: **Logistic regression**, **Random Trees**, **XGBoost Tree**, **CHAID**, and **C&R Tree**). Note: you can tweak the settings of an algorithm in the Settings column (and actually run multiple variations of an algorithm). You will keep the default.

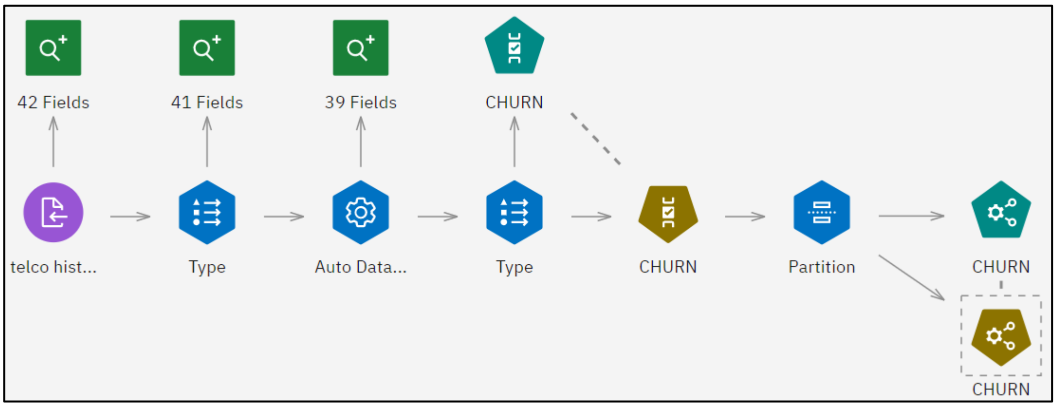
   Note: In data science, a model is often regarded as a "black box". That is, many data scientists just consume models without knowing the details of what's going on behind the scenes. They focus on the accuracy of the model or another evaluation metric. The only requirement is that a model does not perform nonsense computations such as computing the mean postal code behind the scenes. That requirement is met here because you have set the measurement levels correctly previously.

8. Click **Save**, to return to the flow.

9. Run the **Auto Classifier** node.

   It may take a minute to complete execution.

   The results appear similar to the following:

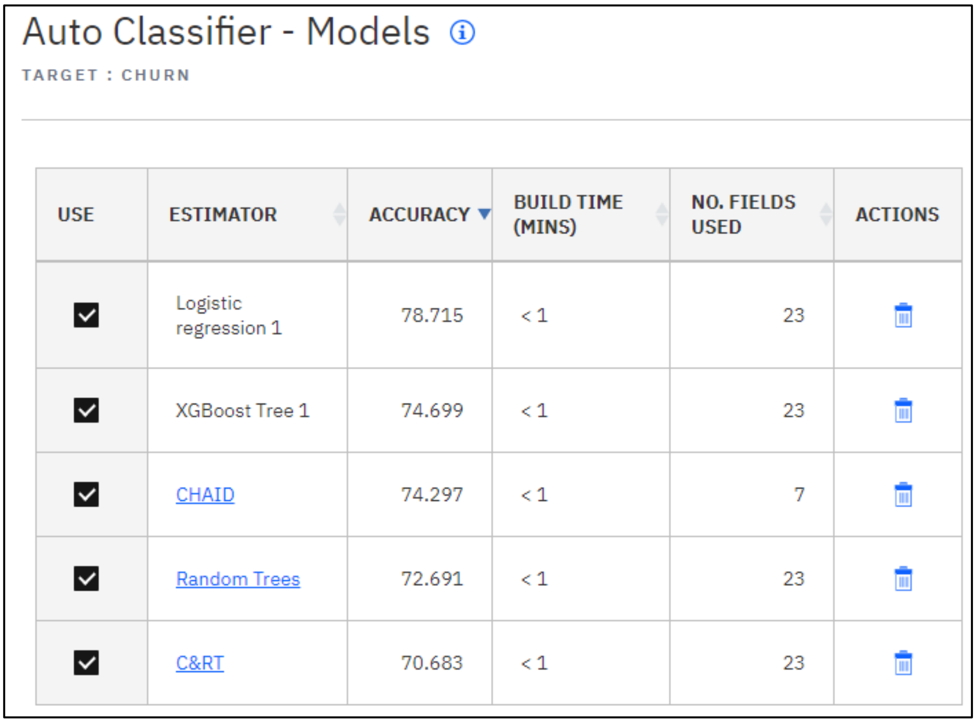


   As before, running an algorithm has generated a model nugget.

8.2 Examine the results in the Auto Classifier model nugget.

1. Click the three vertical dots at the top right of the **Auto Classifier model nugget**, and then click **View Model.**

   The results appear as follows:



Auto Classifier - Models

TARGET : CHURN

USE	ESTIMATOR	ACCURACY	BUILD TIME (MINS)	NO. FIELDS USED	ACTIONS
☑	Logistic regression 1	78.715	< 1	23	🗑
☑	XGBoost Tree 1	74.699	< 1	23	🗑
☑	CHAID	74.297	< 1	7	🗑
☑	Random Trees	72.691	< 1	23	🗑
☑	C&RT	70.683	< 1	23	🗑

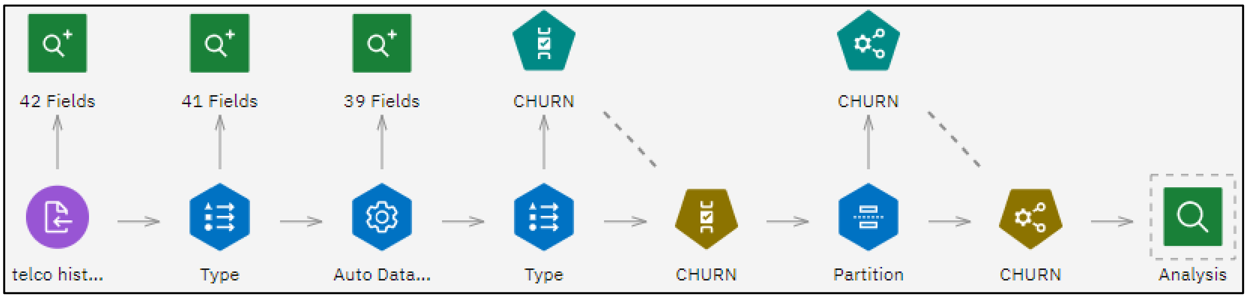
   The best model is Logistic regression, with an accuracy of 78.7%.

   In the Use column, all boxes are checked. This means that the predictions from these five models will be combined into one, single prediction, which makes another model. You can think of this combined model as performing majority voting. If three models predict that a certain customer will cancel and the other two models predict that that customer will stay, the prediction from the combined model is that that customer will cancel.

   Just as you can compute the overall accuracy for an individual model, you can compute it for the combined or "ensemble" model. The Analysis node will compute the accuracy.

2. Return to the flow.
3. Ensure that the **Auto Classifier model nugget** has focus.
4. From the **Outputs** palette, add an **Analysis** node to the flow, and then reposition the nodes somewhat.

   The results appear similar to the following:



5. Run the **Analysis** node.
6. Open the **Analysis** output item.

   The accuracy for the combined model, computed on the testing set, is 77.1 %, which is less than the accuracy of the best individual model, which was logistic regression (78.7% accurate).

   Thus, the Logistic regression model is the preferred model to score new data.
7. Return to the flow.

   There are a few ways in which you can generate a Logistic regression model nugget. Generally, you would add a Logistic node to the flow (downstream from the Partition node), run the Logistic node, and that would generate a Logistic model nugget. Alternatively, in this specific case, you can open the Auto Classifier model nugget and deselect all models except for Logistic regression. That will effectively apply the Logistic regression model only. That is the method that you will choose here.
8. Click the t**hree vertical dots** at the top right of the **Auto Classifier model nugget**, and then click **View Model**.
9. In the **Use** column, deselect **XGBoost Tree**, **CHAID**, **Random Trees**, and **C&RT**.
10. Return to the flow.
11. Run the **Analysis** node.
12. Open the **Analysis** output item.

    The accuracy on the testing set is 78.7%, in agreement with the accuracy of Logistic regression in the Auto Classifier model nugget. That validates that the Auto Classifier model nugget actually represents the Logistic model only.
13. Return to the flow.

That completes this workshop. You evaluated several models in a single run and decided that Logistic regression is the best model to score new data.

9 Score current customers

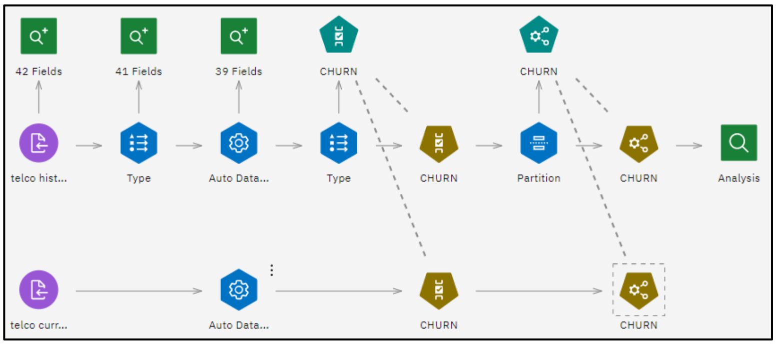
Previously, you have imported the data of 100 current customers (telco current data.csv). In this workshop, you will identify customers who are likely to cancel (and then, for example, you can hand over a list of customers at risk to the marketing department).

9.1 Score new data.

To apply the model to new data, you first need to apply the same transformations to the new data as you did when building the model.

1. Copy the **Auto Data ...** node, paste it, and then add it downstream from the **telco curr...**node.
2. Copy the **Feature Selection model nugget**, paste it, and then add it downstream from the **Auto Data ...** node.
3. Copy the **Auto Classifier model nugget**, paste it, and then add it downstream from the **Feature Selection model nugget** node.

   The results appear similar to the following:



   The flow imports the data of the current customers, automatically prepares it (the Auto Data Prep node), selects fields (the Feature Selection model nugget) and applies the Logistic regression model (the Auto Classifier model nugget, with only Logistic regression enabled).

   Please give it a moment thought why you don't need the Partition node. Also, think about why it is not necessary to include the Feature Selection model nugget as we did here.

   Well, you do not need the Partition node because that is only required when building the model. Here, we do not build a model but apply an existing model to new data, so partitioning the data is not relevant.

   The Feature Selection model nugget is used prior to building a model. It selects the features that are most strongly associated to the target, which is useful to speed up the execution time of building models. Again, here we do not build a model but apply an existing model. If we would not include the Feature Selection model nugget in the lower branch, we would have more fields in the dataset than required to apply the model, but that would not issue an error message.

4. Preview the data at the **Auto Classifier model nugget.**
5. Scroll all the way to the right in the **Data preview** window, so you can view the **$XF-CHURN** field.

   $XF-CHURN stores the predictions from the Auto Classifier model nugget. $XFC-CHURN stores the confidence for that prediction. (Fields added by a nugget are prefixed so that you know what model created the fields.)

6. Validate that the fourth customer is predicted to cancel.
7. Close the **Data preview** window.

That completes this workshop and course. You have scored new customers.

If you have time left, select those who are predicted to cancel, and export their data to a csv file (so that, for example, these customers can be targeted in a campaign).

We hope you have found this workshop useful; thank you for participating!

(Please complete the survey before logging out)