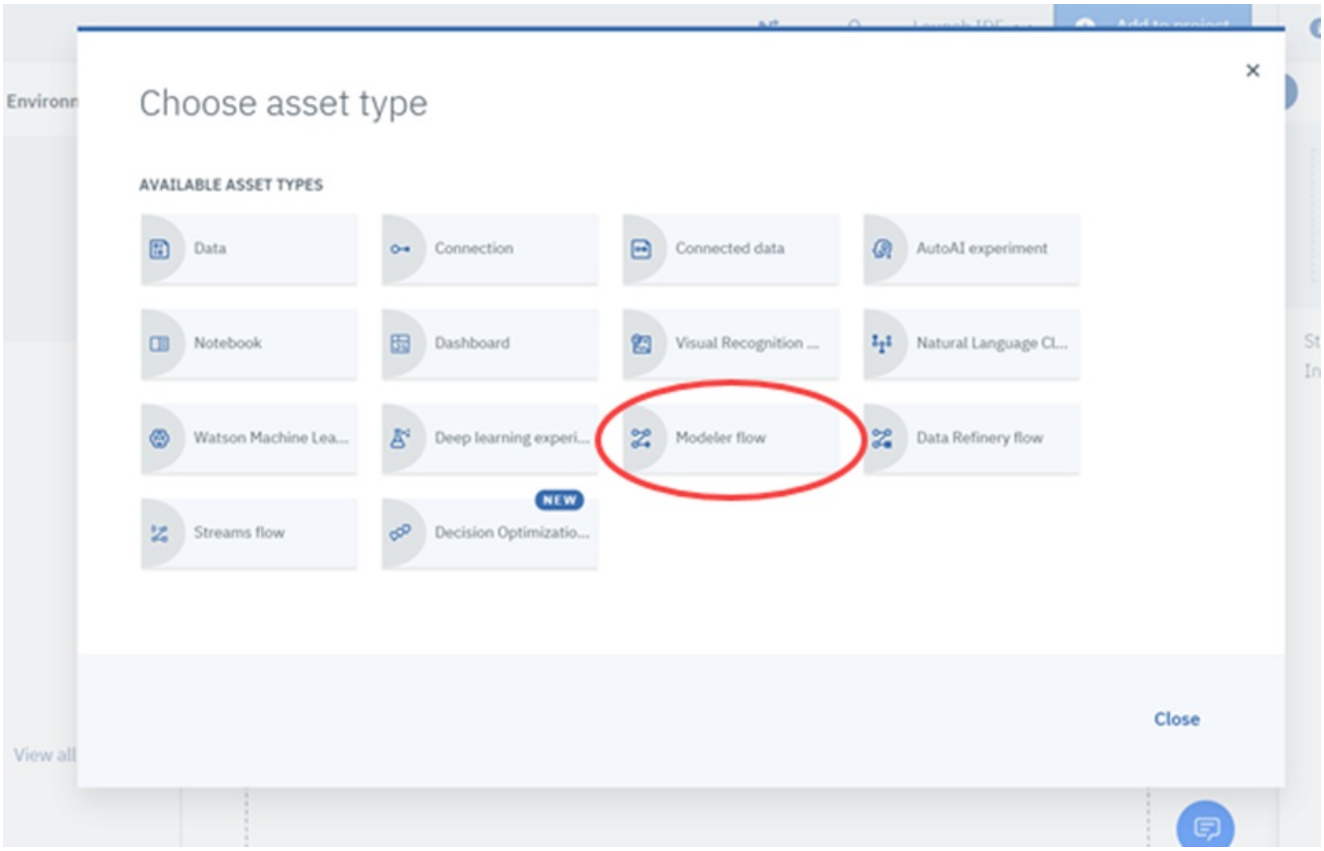
Lab: Modeler Flows in IBM Watson Studio

For this lab, you will use the IBM Cloud Account and the Watson Studio project you created when you worked on the Lab in the previous lesson titled *Creating a Watson Studio Project with Jupyter Notebooks*. To create a free Watson Studio account, use the following link: Watson Studio.

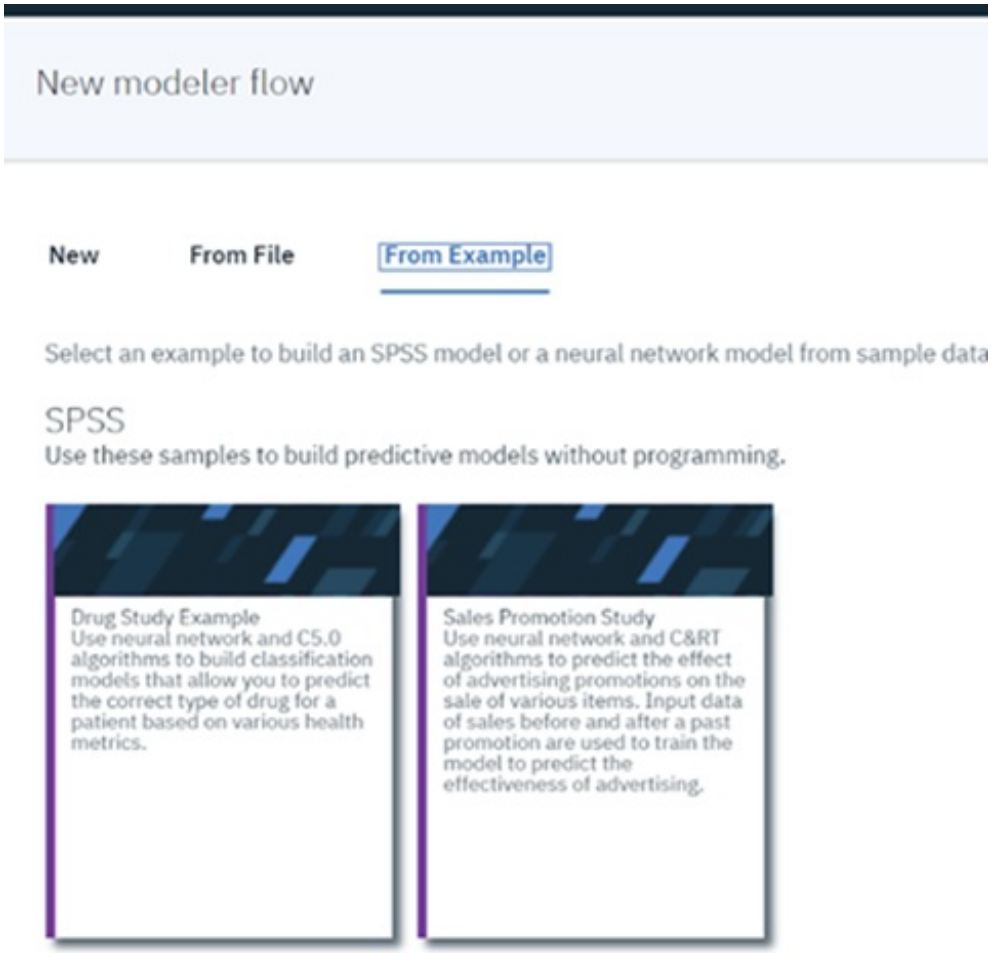
In this lab we will be doing the following:

1. Load an example flow, run it, and examine results.
2. Add an Auto Numeric model, run it, and examine results.
3. Get predictions for new cases using a model we built.

Step 1 - Open your project in Watson Studio, then click "Add Assets" by clicking the blue button on the top of the screen. In the panel that appears, select the "Modeler flow" option.

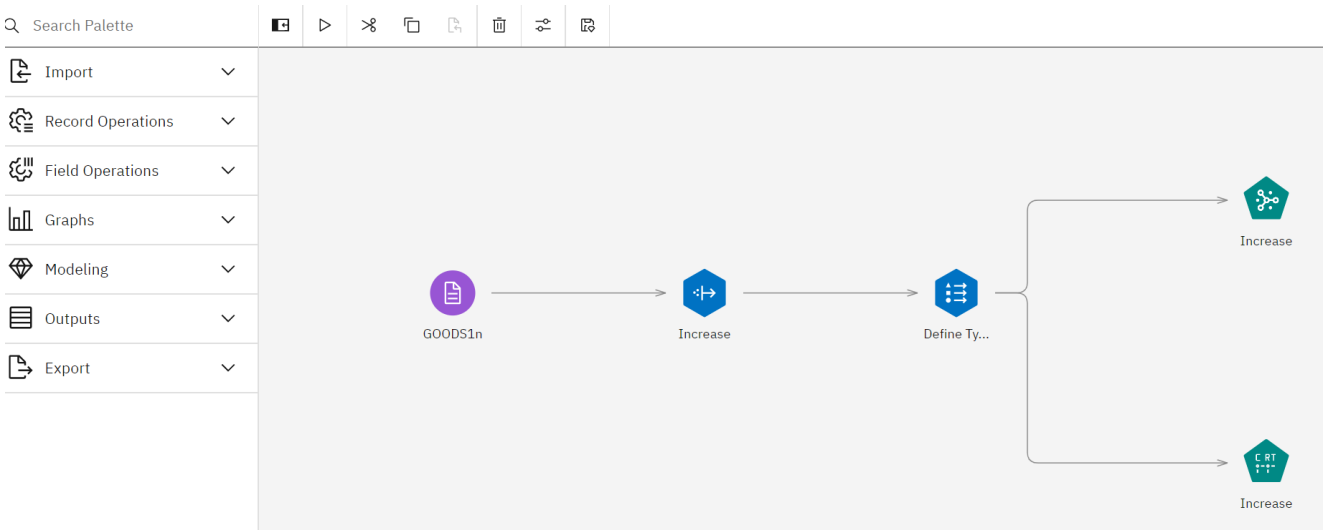


Step 2 - Next, select the tab "From example":



Step 3 - Selecting the Drug study example on the left would give us the flow we have already seen in the previous section, so let's pick the one on the right - Sales Promotion study, then click "Create" button in the lower right corner.

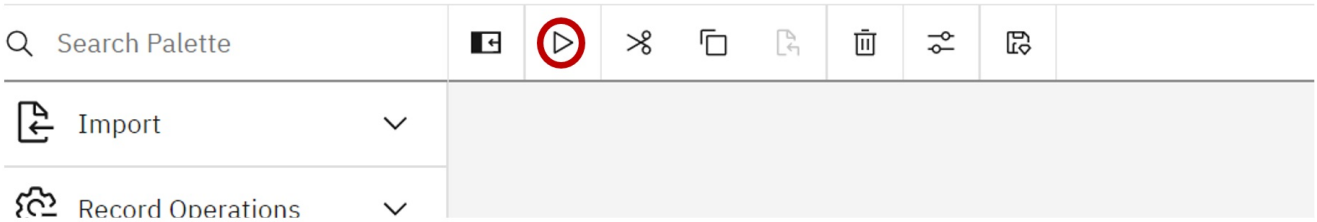
Once the flow loads, you will see this:



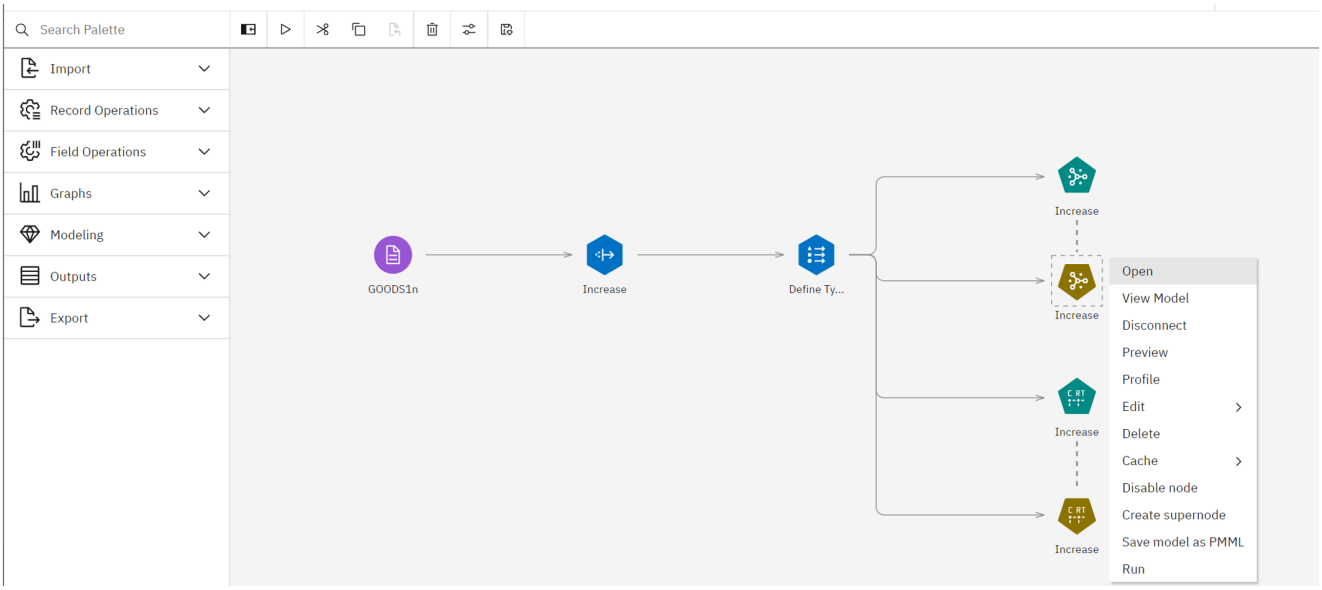
Step 4 - Reading from left to right, we can examine the flow. The purple circle corresponds to the data set of some fictional sales, the next node is deriving a field *Increase* that will be

our target variable. It is based on the increase of sales. Then we see a type node which specifies our target variable and predictors. In this case our target variable is numeric and continuous. Finally, there are two models - a neural network and a C&RT decision tree. We can run the flow as is, or we can modify it to see other possibilities.

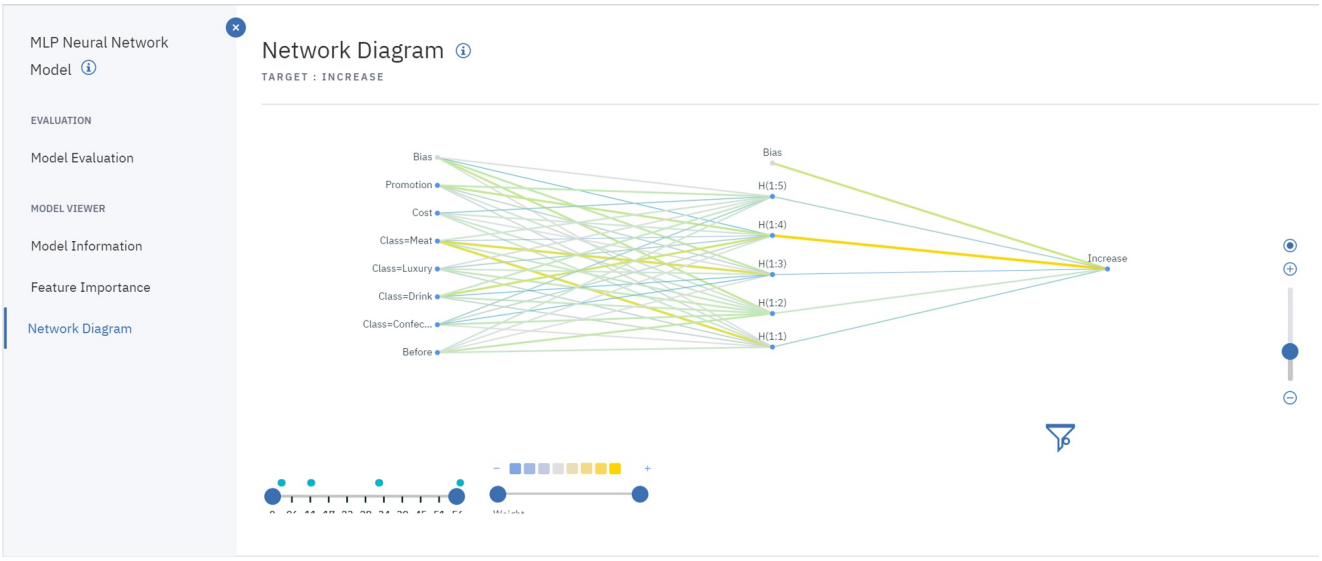
Step 5 - First, let's see what we can get with the existing models. Press the triangular **Run** button on the top of the canvas:



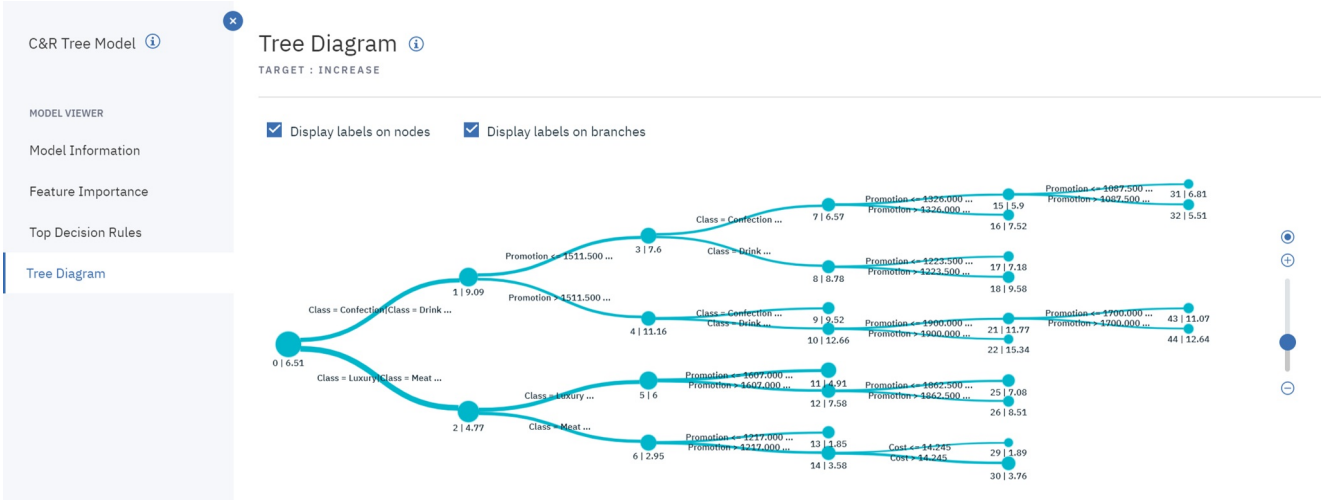
Step 6 - After some execution time, you'll get dark gold model nuggets for the two models. Clicking on three dots in the right side of one of them you can select "View model" and examine the information.



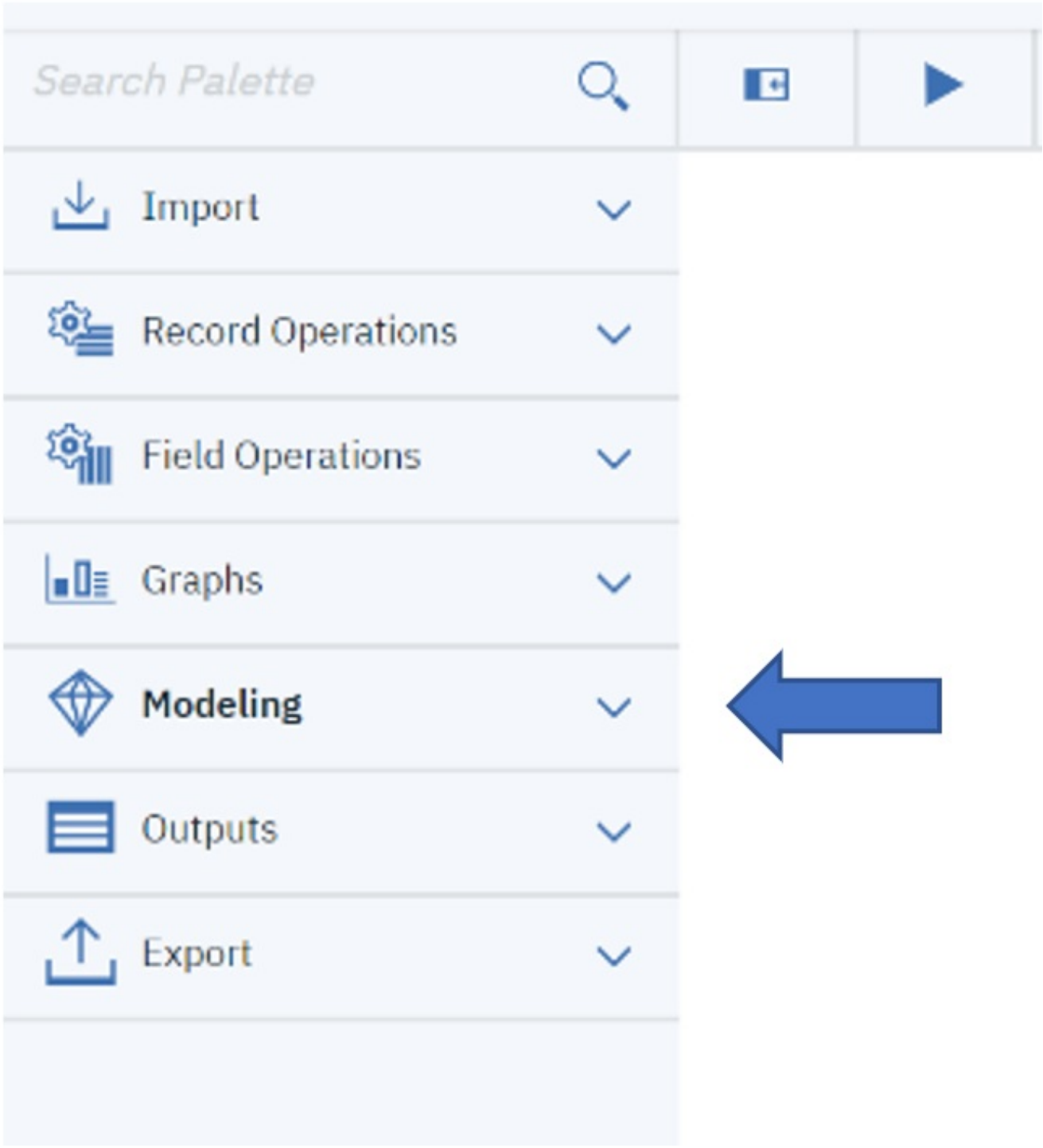
For example, the neural network diagram looks like this:



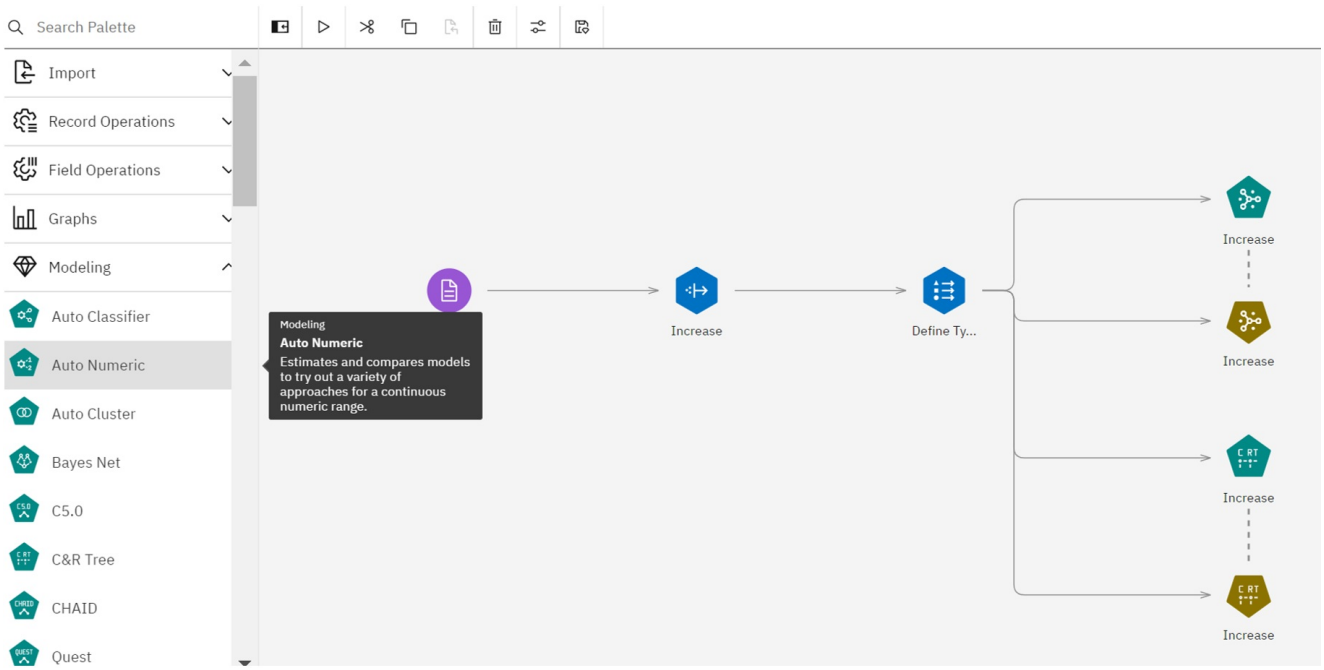
And the decision tree looks like this:



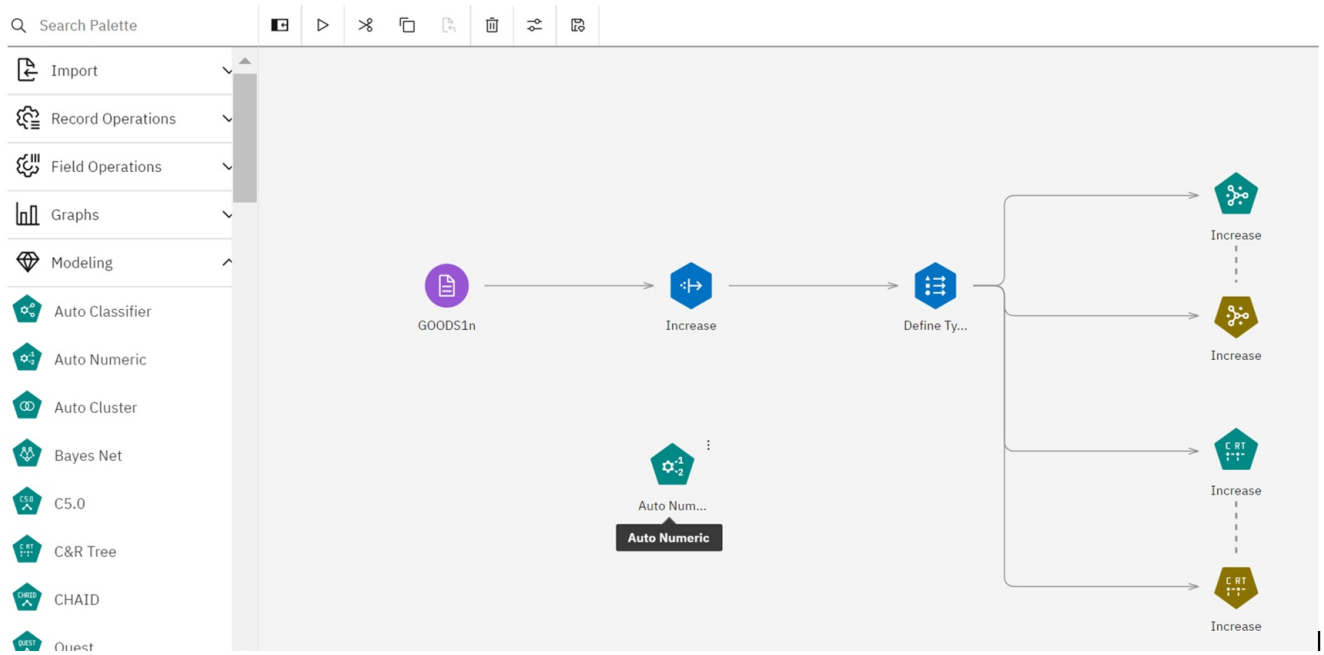
Step 7 - Now, let's add the "Auto-numeric" model to the flow. Open the modeling palette on the left side by clicking on Modeling:



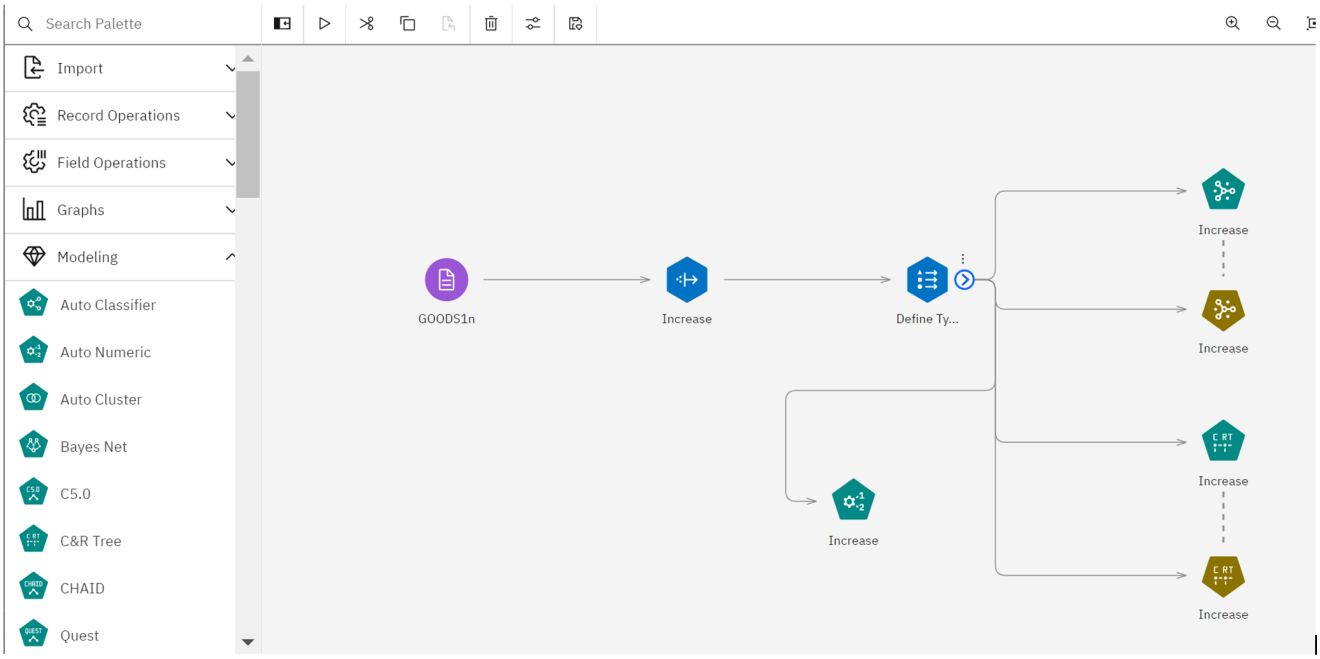
Step 8 - In the resulting modeling palette, pick the Auto-Numeric model:



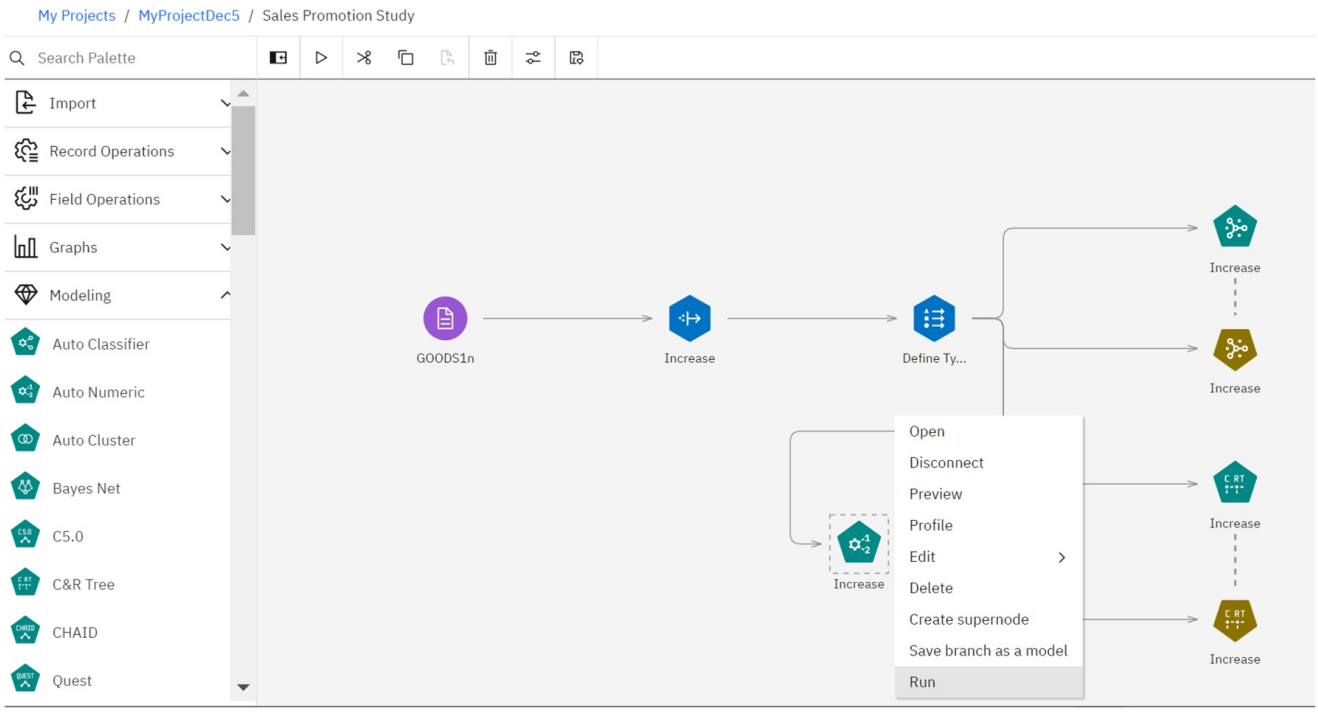
Step 9 - Click on the Auto Numeric node and while keeping the left mouse button pressed, drag it onto the canvas.



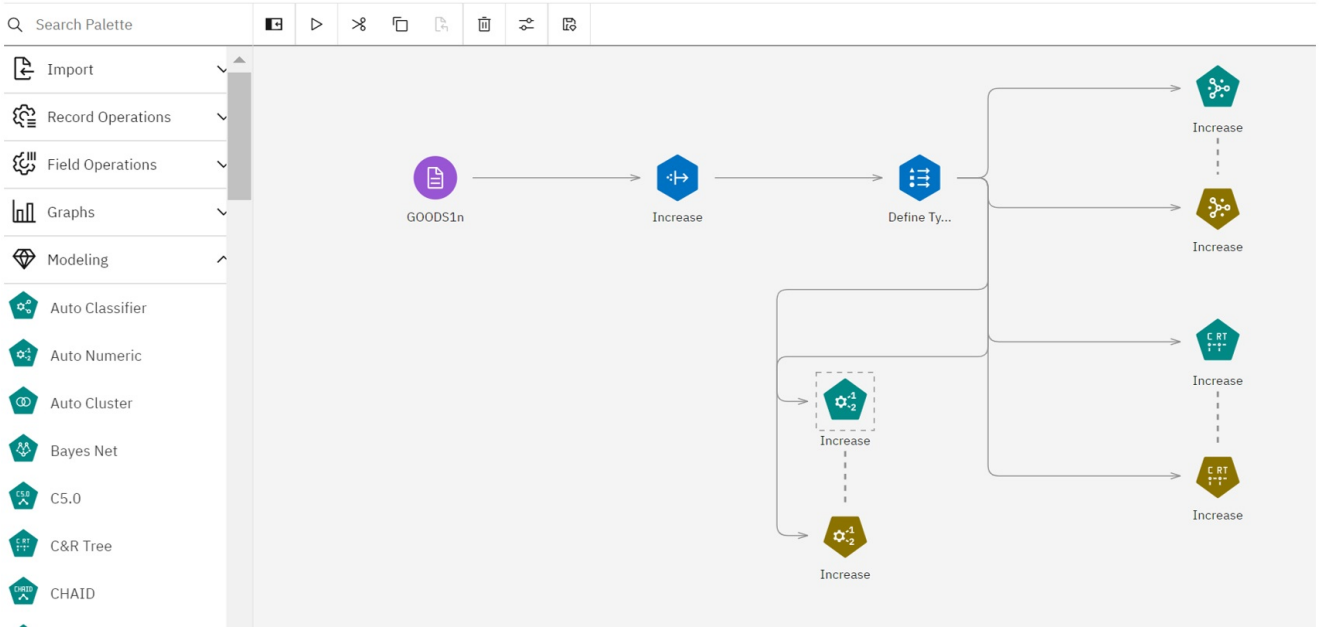
Step 10 - Now, hover over the type node with your mouse and click on the circle with an arrow that appears on the right side. Make a connection to the dot on the left side of the Auto Numeric node.



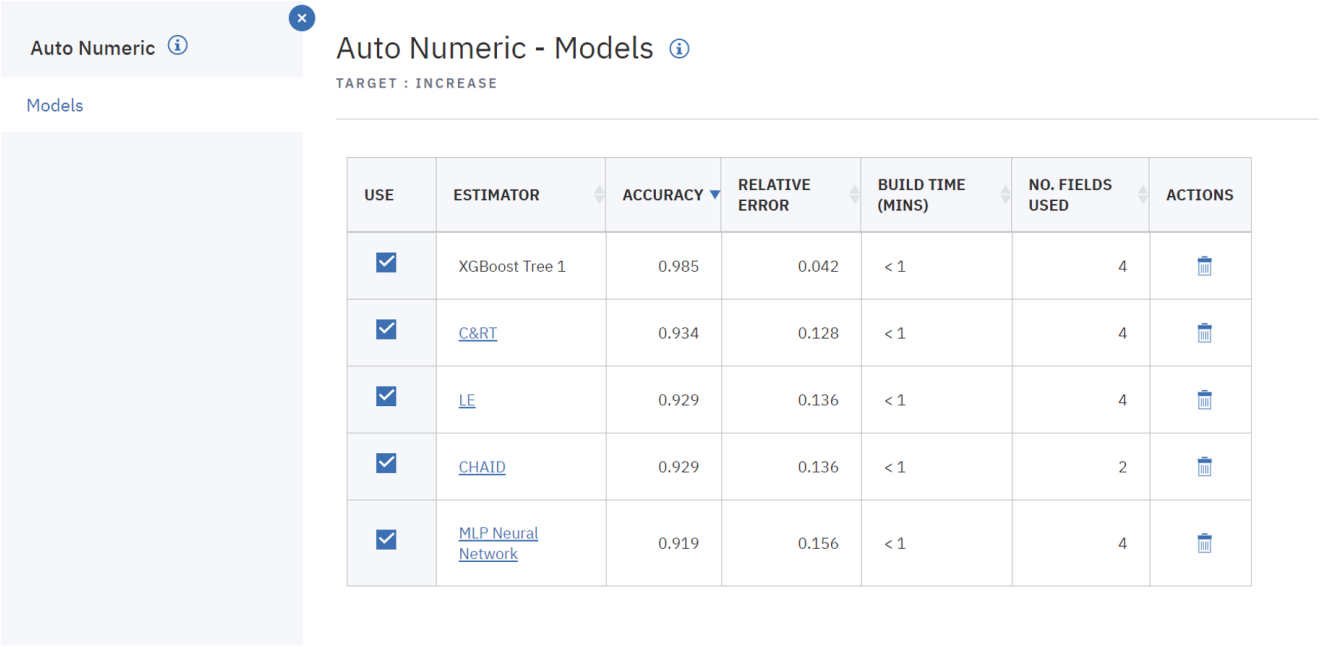
Step 11 - Finally, run the new branch of the flow by clicking on the three dots on the upper right part of Auto Numeric node and selecting the last option in the menu, "Run".



Step 12 - After the execution ends, we get a model nugget:



and viewing the model, we see the following table:

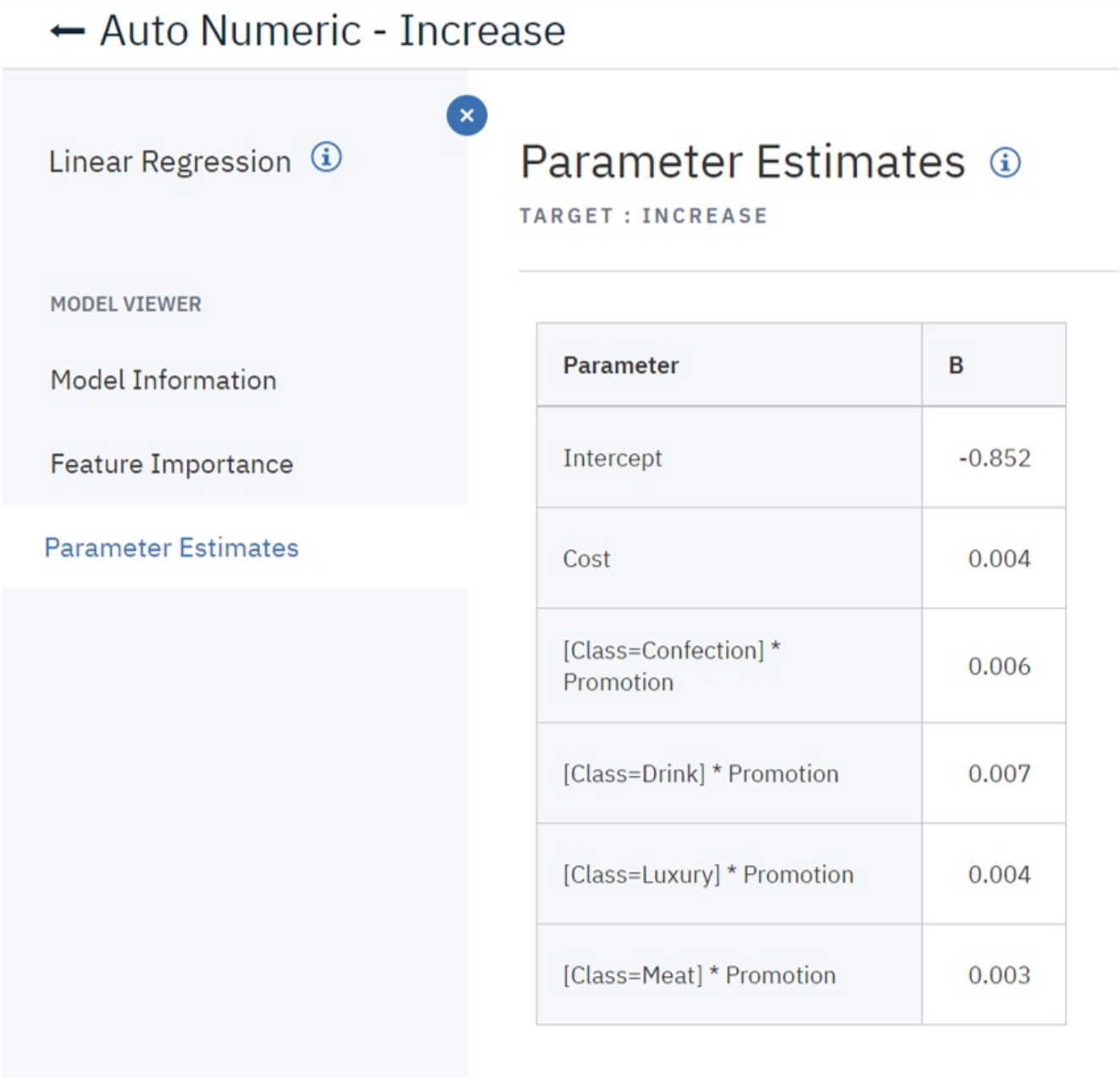


USE	ESTIMATOR	ACCURACY	RELATIVE ERROR	BUILD TIME (MINS)	NO. FIELDS USED	ACTIONS
☑	XGBoost Tree 1	0.985	0.042	< 1	4	
☑	C&RT	0.934	0.128	< 1	4	
☑	LE	0.929	0.136	< 1	4	
☑	CHAID	0.929	0.136	< 1	2	
☑	MLP Neural Network	0.919	0.156	< 1	4	

This tells us that five different models have been built, and also some properties of the models.

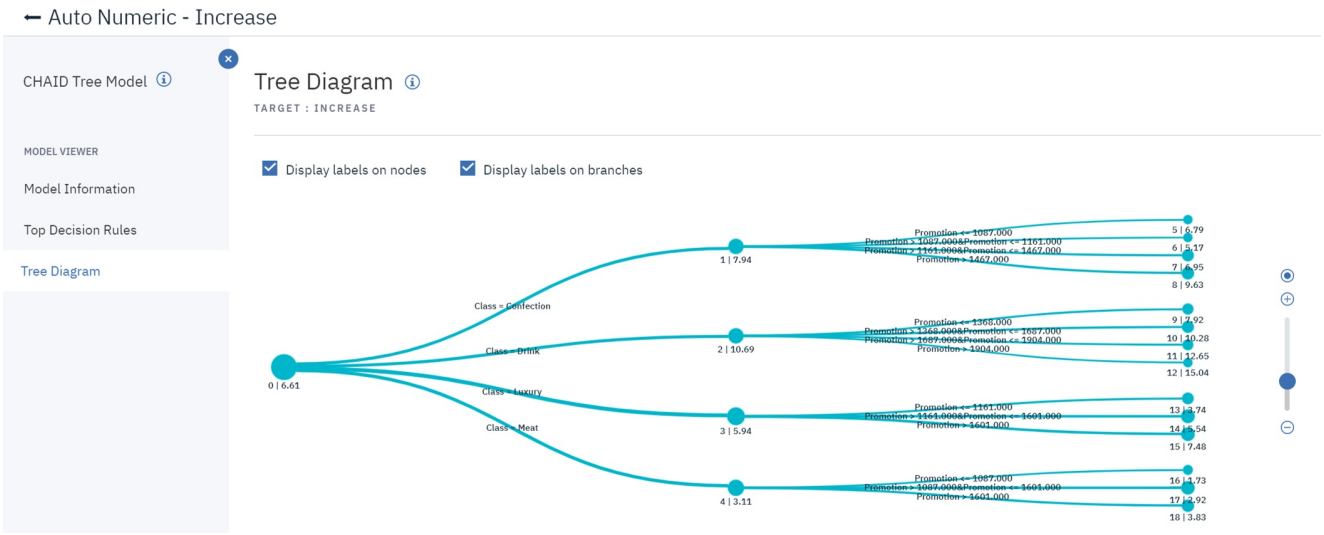
1. XGBost is a very popular model, representing gradient-boosted ensemble of decision trees. The algorithm was discovered relatively recently and has been used in many solutions and winning data science competitions. In this case, it created a model with the highest accuracy, which "won" as well.
2. "C&RT" stands for Classification and Regression Tree", a decision tree algorithm that is widely used. This is the same decision tree we saw earlier when we built it separately.

3. "LE" is "linear engine", an IBM implementation of linear regression model that includes automatic interaction detection. The model coefficients are shown in **Parameter Estimates** table. We can see that several coefficients correspond to a combination of one category of variable *class* with continuous variable *Promotion*. This is called "an interaction effect". You will not see such features in simple linear regression models.



Next, in the table of models was CHAID. It is another algorithm for building decision trees. The acronym **CHAID** stands for Chi-squared Automatic Interaction Detector. It is one of the oldest tree classification methods originally proposed by Gordon Kass in 1980. Clicking on the algorithm name we can see the details of the that model. Most decision tree algorithms (including C&RT) build binary trees, i.e. trees where each node has either zero or two children. CHAID is different, it produces trees with multiple children for some

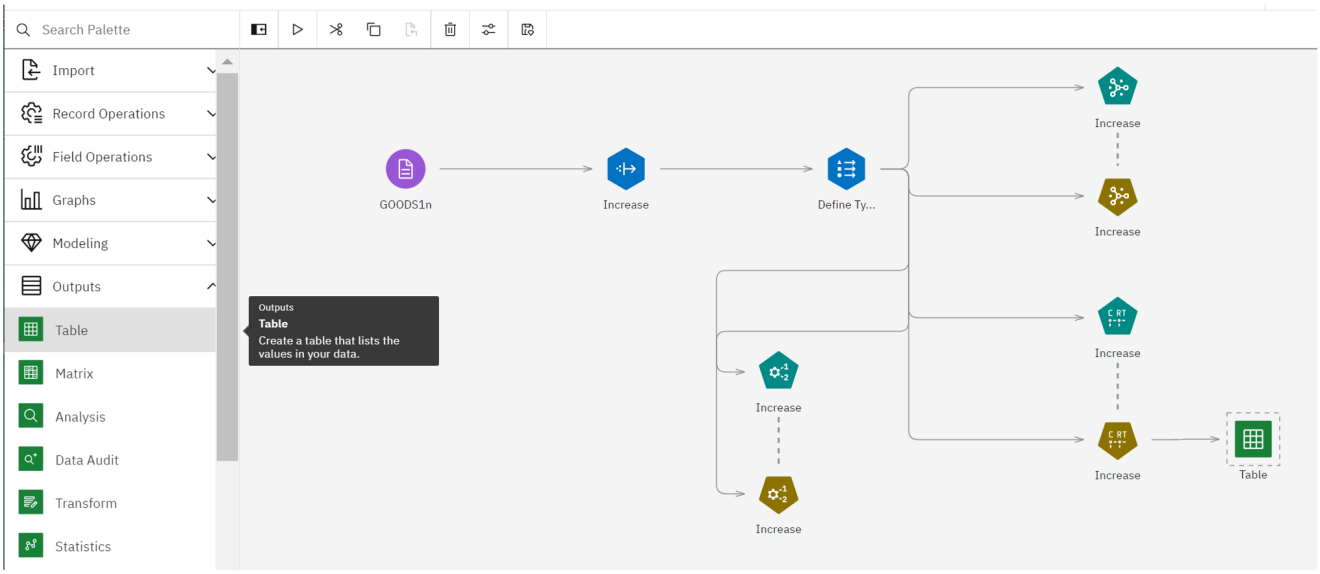
nodes. Such trees can be not so deep but very wide. We see an example here. The root node has four children, with the split based on *class* variable. Each child node has three or four children of its own, all split based on *Promotion*. So essentially, we have an interaction of *Class* and *Promotion* in this model as well.



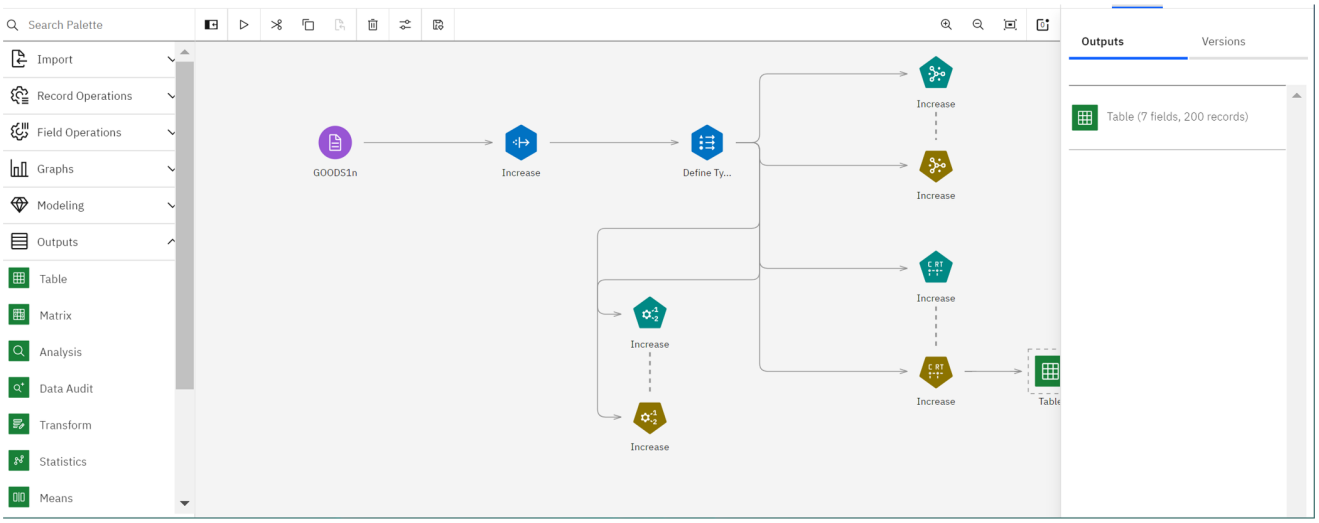
Finally, the last model built by the Auto Numeric node is "MLP Neural Network", it is the same neural network as in the original stream. "MLP" stands for "multi-layer perceptron", a name used for the fully connected feed-forward neural networks popular in the 1990's.

Step 13 - How can we use a model we built to get predictions for new data (or perhaps the data we already used)? In Modeler flows it only requires a few steps.

Create a data source node for the new data, attach a type node, connect that to the model nugget, and add a "table" node after the model nugget. In our example we will just get predictions for the original data using the C&RT model, so we open the "Output" group on the node palette, drag the "Table" node and add the connection to it from the model nugget:



Step 14 - Now run that new branch, and get a table node on the right:



Double-clicking on that node we can see the data with added new column **$R-Increase** (containing predictions):

Class	Cost	Promotion	Before	After	Increase	$R-Increase
Confection	23.990	1467	114957	122762	6.789	7.522
Drink	79.290	1745	123378	137097	11.119	12.640
Luxury	81.990	1426	135246	141172	4.382	4.908
Confection	74.180	1098	231389	244456	5.647	5.510
Confection	90.090	1968	235648	261940	11.157	9.525
Meat	69.850	1486	148885	156232	4.935	3.761
Meat	100.150	1248	123760	128441	3.782	3.761
Luxury	21.010	1364	251072	268134	6.796	4.908
Luxury	87.320	1585	287043	310857	8.296	4.908
Drink	26.580	1835	240805	272863	13.313	12.640
Drink	65.230	1194	212406	227836	7.264	7.175

Step 15 - Now you can experiment with Modeler flows on your own. You can use various nodes, just don't forget to put a Type node before any modeling node.

Some other possible models that can be used to predict a continuous target are: Genlin, GLMM, LSVM, Regression, KNN, XGBoost Linear. If your data has a categorical target, you may want to try Auto Classifier, as well as C5, C&RT, CHAID, Logistic Regression, SVM, Neural Networks, and many other models.

You will learn more about various models when you take a course on Machine Learning.