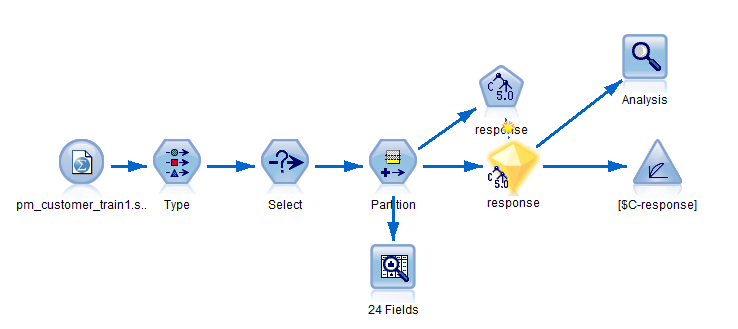
**Modeling Customer Response (Decision Tree)**

The example in this note is created to demonstrate how to evaluate the effectiveness of a model as well as to show a process that is commonly used in practice. You will learn two new metrics -***Gain***and ***Lift*-** that are useful in practice.

This example is based on a fictional company that wants to achieve more profitable results in future marketing campaigns by matching the right offer to each customer. Specifically, the example uses a Decision Tree model to identify the characteristics of customers who are most likely to respond favorably, based on previous promotions, and to generate a mailing list (calling list, email etc.) based on the results. This same process can be applied in many other business scenarios like, acquiring new customers, running a campaign for donations etc.

In the application, we will first build a classifier. A decision tree is used in the stream below, but any other classifier may also be used. The Decision Tree algorithm can generate rules that indicate a higher or lower likelihood of a given binary (yes or no, *responder* or *non-responder*) outcome.

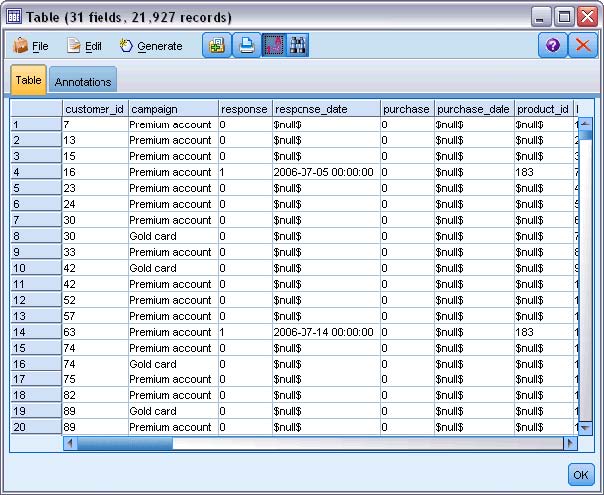
*Decision Tree sample stream*



The data file *pm\_customer\_train1.sav* is available on the Black board.

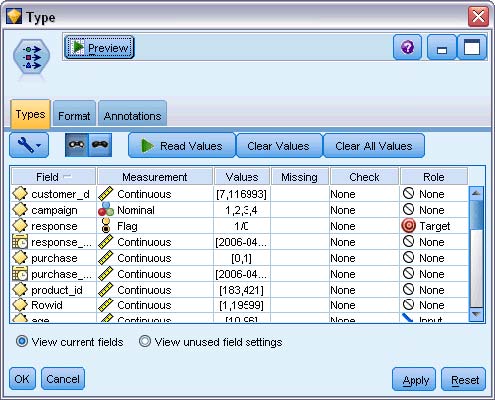
***Historical Data***

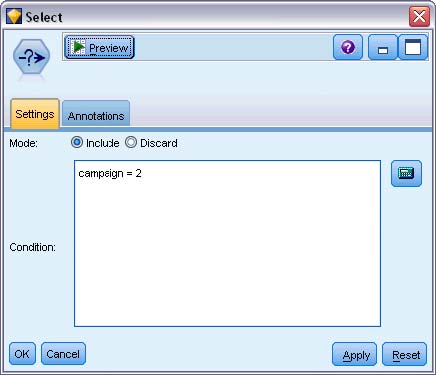
The file *pm\_customer\_train1.sav* has historical data tracking the offers made to specific customers in past campaigns, as indicated by the value of the *campaign* field. The largest number of records fall under the *Premium account* campaign.

The values of the *campaign* field are actually coded as integers in the data, with labels defined in the Type node (for example, *2 = Premium account*). You can toggle display of value labels in the table using the toolbar. The file also includes a number of fields containing demographic and financial information about each customer that can be used to build or “train” a model that predicts response rates for different groups based on specific characteristics.

***Building the Stream***

* Add a Statistics File node pointing to *pm\_customer\_train1.sav*, wherever it may be located on your computer. Read in the values. Add a **table** node from output to browse through the data to verify that the data is read properly.
* Add a Type node, and select ***response***as the target field (Role = **Target**). Set the measurement level for this field to **Flag.**
* Set the role to **None** for the following fields: *customer\_id*, *campaign*, *response\_date*, *purchase*, *purchase\_date*, *product\_id*, *Rowid*,and *X\_random*.
* These fields all have uses in the data but will not be used in building the actual model.

 Click the **Read Values** button in the Type node to make sure that values are instantiated.

Although the data includes information about four different campaigns, you will focus the analysis on one campaign at a time. Since the largest number of records fall under the Premium campaign (coded *campaign = 2* in the data), you can use a Select node to include only these records in the stream.

***Creating the Model***

* Attach a Decision Tree node to the stream and build a Model.
* Once model is built, as usual attach an analysis node. Also attach an Evaluation node from the Graphs Palette.
* Edit the Evaluation node and select “Gains” chart type.
* Next select Cumulative Plot, Include baseline, and also include best line.
* Using Expression builder, you need to specify that the Gain chart or Lift chart that

You want are for the @TARGET = 1.0, as show below.

* Click RUN and it will create the Cumulative Gains Chart.
* Similarly, you can create a Lift chart as well.

