Creating a Probabilistic Model in Informatica Data Quality
Abstract

This article describes how to create a probabilistic model in Informatica Data Quality.

Supported Versions

- Informatica Data Quality 9.5.0 and later

Table of Contents

Creating a Probabilistic Model in Informatica Data Quality ............................................................. 1
Overview .......................................................................................................................................... 2
Why NLP .......................................................................................................................................... 2
Probabilistic Model Structure ........................................................................................................... 3
Probabilistic Model Reference Data ................................................................................................ 4
Probabilistic Model Label Data ........................................................................................................ 5
Creating a Data Object .................................................................................................................... 5
Creating a Probabilistic Model ......................................................................................................... 6
Using an Untrained Probabilistic Model .......................................................................................... 7
Training the Probabilistic Model ...................................................................................................... 7
Alternative Method to Train the Probabilistic Model ........................................................................ 8
Using a Trained Probabilistic Model ............................................................................................... 9

Overview

A probabilistic model is a reference data object. Use a probabilistic model to understand the contents of a data string that contains multiple data values. A probabilistic model identifies the types of information in each value in the string.

You can add a probabilistic model to a Labeler transformation and a Parser transformation:

- Use a probabilistic model in a Labeler transformation to assign a descriptive label to each value in a data string. The Labeler transformation writes the labels to an output port in the same format as the input string.

- Use a probabilistic model in a Parser transformation to write each value in an input string to a new port. The Parser transformation creates an output port for each data category that you define in the probabilistic model.

Probabilistic models use Natural Language Processes (NLP) to identify the type of information in a string. NLP detect relevant terms in the input string and disregard terms that are not relevant.

You compile a probabilistic model in the Developer tool. When you compile a model, you create associations between similar data values in the model. The Labeler and Parser transformations use the compiled data to analyze the values in the input strings.

Why NLP

Parsing or standardizing works correctly only if a column has a clear identifier such as an address or a company name. It requires an exact match to the reference table entries to make a positive match. For a given input, if these identifiers are not very clear, you need numerous transformations to parse and label the data.
In addition, you need reference tables to identify the known values, token sets for different token types (word, number, and so on), regular expressions for custom data structures and patterns to split data by known patterns and their frequency of occurrence.

To solve this problem, you can use a probabilistic model to predict the relationships between words that allow the labeling of ambiguous data and to reduce the complexity of Data Quality mappings. A probabilistic model also improves performance and gives better results.

**Probabilistic Model Structure**

A probabilistic model contains a column of reference data values and one or more columns of label values. The reference data values represent the different values that can appear in the input data. The labels represent the types of information that the input data values can contain.

A Labeler or Parser transformation uses the label values to analyze the structure of the data in an input row. The structure of the row determines the type of information in each data value. Each input row can have a different structure. When you assign labels to the reference data values, you define the possible order in which the input data values might appear.

A probabilistic model also contains compilation data. The transformations use the compilation data to calculate similarities between the reference data and the input data. When you compile a model, you create or update the compilation data.

The following figure shows a probabilistic model in the Developer tool:
When you use a probabilistic model in a Labeler transformation, the Labeler assigns a label value to each value in the input row. For example, the transformation labels the string "Franklin Delano Roosevelt" as "FIRSTNAME MIDDLENAME LASTNAME."

When you use a probabilistic model in a Parser transformation, the Parser writes each input value to an output port based on the label that matches the value. For example, the Parser writes the string "Franklin Delano Roosevelt" to the FIRSTNAME, MIDDLENAME, and LASTNAME output ports.

### Probabilistic Model Reference Data

The reference data values in a probabilistic model represent the types of input data that a Labeler or Parser transformation might read in a mapping.

You can add, edit, and delete reference data rows in the Developer tool. You can paste reference data from the clipboard. You can also use mapping source data as a source for the reference data values in a probabilistic model.

After you add the data values, create the label columns and assign a label to each data value in each row of the model.
Probabilistic Model Label Data

A probabilistic model contains descriptive labels for the types of information in the reference data. When you create a model or add reference data to a model, assign a label to each reference data value.

The labels you create appear as columns in the probabilistic model. When you assign a label to a data value, the model adds the value to the label column. You can assign any label in the model to any reference data value. If the same value has different meanings in two rows of reference data, you can assign different labels to the value in each row.

You can define the same combination of labels for multiple input strings. Multiple examples of a label increase the likelihood that the probabilistic model assigns the correct label to an input data value.

Creating a Data Object

You can create a probabilistic model from a data object by parsing a sample of the source data file. You can use parsing capabilities such as reference tables and token sets to partially label the data.

1. Import a flat file source as a Physical Data Object (PDO). Import a sample (the first 100 rows) of the file that can be parsed using the probabilistic model. Set the field precision corresponding to the data.
2. Create a new mapping and add the PDO that you created.
3. Using a Case Converter to convert the input string to uppercase.
4. If any punctuation exists, remove the punctuation using a Labeler transformation to label symbols as * and a Standardizer transformation to remove * from the string.
   
   Sample Input String: D. R. Components Keith C Augello Manager 600 W. Hillsboro Blvd.
   
   After removing punctuations: D R Components Keith C Augello Manager 600 W Hillsboro Blvd

5. Add a Labeler transformation to label the data using some of the Accelerator RTMs.
   
   Note: Set the LabeledOutput port precision and TokenizedData corresponding to the input data length.

   The following snippet shows sample data and the configuration details:
   
   Sample Input String: D R Components Keith C Augello Manager 600 W Hillsboro Blvd

   Use the following delimiter and reference tables:

   - Delimiter is space (\s).
   - outputDelimiter is <:>.
   - Reference tables:
     - usa_firstnames_infa
     - usa_surnames_infa
     - usa_occupation_infa
     - usa_company_names_std_infa
6. Create an empty flat file PDO, and add columns for the LabeledOutput and TokenizedData ports. Set the file name as Label_Desc.csv and set a target directory.

7. Add the flat file PDO that you created in step 6 to the mapping as a target. The following figure shows a sample mapping:

8. Save, validate, and run the mapping.

9. Open the target PDO and run Data Viewer on it. The labeled and tokenized data appears.

Creating a Probabilistic Model

You can create a number of probabilistic models and increase the number of rows of trained data in each model to understand how quickly results can be achieved.

1. Create a PDO for the Label_Desc.csv file. This is the target file that was created in the earlier section.

2. Open an existing content set or create a new one.
   • Click the Content tab, select Probabilistic Models, and click Add.
   • Select the Probabilistic Model from Data Objects option and click Next.
   • Enter the name of the model as PM_Untrained. Browse and select the PDO created with Label_Desc.csv and click Next.
   • Select the column with labeled data and click Label >.
   • Select the tokenized data and click Data >.
   • Click Next.
   • To use 30 rows of data, edit the sampling options value to First 30 rows.
   • Change the delimiter for the data to <:>
   • Leave the label for unassigned tokens as O.
   • Click Finish.
The untrained model will have some fields parsed to the labels that you defined, but the majority will not.

3. Save the content set and compile the model so that you can use it in a Parser transformation.

Using an Untrained Probabilistic Model

1. Import the flat file source as a PDO. Set the field precision corresponding to the data length.
2. Create a new mapping and add the flat file PDO.
3. Add a Case Converter to change all data to uppercase.
4. Add a Parser transformation and configure it to parse tokens using probabilistic matching techniques. In the Probabilistic Models dialog box, select the untrained probabilistic model that you created and click Finish.
5. Run Data Viewer on the Parser transformation and review the results.
   
   Note: You can right click the Parser transformation and select Profile Now. Review the profile results. The most common label fields are incorrect or null. Therefore, the untrained model is not very effective.

Training the Probabilistic Model

1. Create a new probabilistic model and name it distinctively since you need to create at least one more model, say PM10. Use the data object that you created with the sampling option set to 10.
   
   Note: Based on the input data that you use, when you review the probabilistic model prior to training it, you will notice that the Un-defined label is the most prevalent, while the other data such as company is poorly defined and LName contains a lot of erroneous data.
2. To train the data, select the label that you want to train. For example, select COMPANY and click the Pencil icon. The tokens that are assigned to COMPANY are highlighted in yellow.
3. Double-click each token that you want to assign to the COMPANY label.
4. Repeat these steps for FName, Lname, and Title. Correct any labels that you feel are incorrect. For example, if ‘Fritz’ is assigned to Lname, the assignment might be incorrect because the data context suggests that it should be assigned to FName.
5. Add an INIT label because some names will also include an initial.
7. Add a new label, name it as INIT, and click OK.
8. Assign tokens to the INIT label.
9. Assign any address elements of the data to the O(ther) label.
Alternative Method to Train the Probabilistic Model

As you need to train every row and edit many labels, you can perform the following steps to quickly train the probabilistic model:

1. Select the first row of data.
2. Expand the **Labels** option at the bottom of the **Content Set** tab.
3. Edit the labels by selecting different labels from the drop-down list.
   
   **Note:** You can also select the first row label and press the down arrow key to highlight the next row.

   Press the first letter of the label that you want to assign twice. For example, if you press C twice, the COMPANY label is assigned. You can press Enter and then press the down arrow key to move to the next row.

   After you assign all the data to labels, the U(un-defined) label should be empty.

   To remove the U label, select the **Edit** option on the **Labels** menu, select **U**, and then click **Delete**.

4. Save the content set and compile the model.
Using a Trained Probabilistic Model

1. Open the mapping that was used to test the untrained model.
2. Add a second Parser transformation and configure it to parse tokens using probabilistic matching techniques. In the Probabilistic Models dialog box, select the PM10 model, and click Finish.
3. Run Data Viewer on the Parser transformation and review the results.
   - Overflow should contain a lot less data than the Parser transformation when you use the untrained model.
   - Labels are populated for more rows.
4. Right-click the Parser transformation and select Profile Now. Review the profile results.
Informatica recommends using more data to train the model. To demonstrate how quickly we can improve results, repeat the tasks described under the Training the Probabilistic Model section for regular samples say 20, 40, 60, and so on. Apply the model by repeating the tasks described under the Using a Trained Probabilistic Model section. Compare the profiles of the Parser transformation using the PM10 model and the newly created PM20 model. Note the improvement in parsing.

Author
Lakshminarayanan Arunachalam
Technical Support Engineer