**1-5– Wild Matches & Grouping**

Errors and typos are common when working with data. It can be due to human error - by typing in names and addresses in wrong spelling, and can also be caused by sourcing from different databases-wherein values might be coded and structured differently. Having such discrepancies makes it problematic when it comes to matching records and connecting the trail of data from one table to another. You might easily find the name “Oasis Inc” on your customer table, but can’t join it to your transactions table just because it has the “Incorporated” suffix is not abbreviated. The Fuzzy Match tool is developed for such cases.

**Fuzzy Match tool is used to identify non-identical duplicates of a dataset by specifying match fields and similarity thresholds.**

For this example, we are going to match 2 small datasets from 2 text inputs. The first input would be the “Company Master”, this is the original list that contains the unique company names. The 2nd input called “List” has the set of company names which has duplicates and typos. Both text inputs contains the column “Source” which tags the datasource name of each set. We will use the fuzzy match tool to find the original names in our company master that closely matches the company on the list. From the Join tool tab, drag Fuzzy Match and place it in the canvas. Fuzzy Match has a single input anchor so we still need to prepare and union our data. One of the fields required when using fuzzy match is a unique ID to identify the records. Drag Record ID from the preparation tool tab and connect it after the first Text Input. Set the ID name to ListID with starting value “1”. Next, add another Record ID to the canvas and connect it to the 2nd Text Input. Set the ID name to ListID but this time set the value to start at “99”. Once done, add a union tool from the Join Tool Tab and connect it to the output anchor of the 2 Record ID tools. Now that we have the data on a single stream, connect the output anchor of the union tool to the fuzzy match tool.

**From its configuration, fuzzy match has 2 modes. You can choose to either Purge or Merge. Purge is used if the data you want to match only comes from a single source. So, if you wanted to identify duplicates of customers, companies, or products in a single table, you can apply this tool to identify wild matches. On the other hand, Merge is used if you need to match records from different sources, like the 2 text inputs in this example. Select the radio button for merge mode to activate the rest of the configuration.** When using merge, you need to map the identifier for the source. For example, our text inputs have the field “Source” in order for alteryx to identify which records belong to which dataset. Select “Source” on the dropdown of the source ID to set it. If your own file does not have its source tag, you can choose “Output File Name as Field” in the Input Data Tool to use the filename or path as your source ID. If you are not using a file as input, simply use the formula tool to create a tag for the source ID. Back to our fuzzy match, next we need to map the Record ID field. From the dropdown select the “ListID” that we created from the Record ID tool earlier. Below this option is the Match Threshold which lets you set a percentage for cutoff. A higher percentage means the match will be stricter. This will depend on the match score of each paired record. The Match score takes into consideration each specification within the configuration properties of the Fuzzy Match tool: Each field, the match style, the match weight, and the resulting field match score is considered in calculating the score, which is then against the specified Match Threshold. Set the threshold to 70%. After the threshold is the Match Fields where you can set the field name to match on, and its match style. On the field name, set “Company Name” then select the Match Style “Company Name”. There are also a bunch of predefined match style available such as Address, Name, and Zip. You can also choose the style “Custom” to set your own match rules. Custom can be configured by pressing “Edit”. **Preprocess sets the cleaning procedures that will be done before the Generate Keys & the fuzzy match function. Generate Keys is the method by which a potential match is identified. Alteryx read through the match fields, and assigns keys based on the components of the field. A potential match is found if the keys generated are equal for 2 records. Finally, Match Function is a more granular process by which a match is identified, and a score is applied. This is different from the keys process which must match exactly. Match Function lets you choose from a combination of Levenshtein Distance and Jaro Distance when looking for a match. Since we are using the predefined “Company Name” match style, the configuration is set to Double Metaphone for generation of Keys and Jaro Distance on the match function by word.**

**Aside from the Matching configuration, Fuzzy match tool has other advanced options to tweak which fields can be included in the output. Output Match Score presents the match score for each record by appending an additional field. Output Generated Keys shows the key assigned from each match and appends it as a new field. Output Unmatched Records shows the other records that either had no matches, or had a match but did not reach the Match Threshold. Don't Compare Records already in a Group will not match a record that has already been matched to a different row. This reduces processing efforts and time. Generate Keys Only from the functions name itself, it will only output the generated key, and no matching will take place. Check the “Output Match Score” and “Output Generated Keys” then run the workflow.**

As we can observe from the Output, 5 fields were generated. ListID for the ID from Company Master, ListID2 for the ID from the list, MatchScore and MatchScore\_Company Name for the scores, and MatchKey for the unique generated key on each match. Since the output does not show the company name, we can just join it back to the original data stream. From the Join tool tab, Drag Join tool into the canvas. Connect the left anchor to the output of the fuzzy match, then connect the record ID tool output of the company master to the right anchor of the join tool. Set it to join by specific fields using the column ListID for both left and right. You can also uncheck the ListID of the Right datastream to prevent duplicate columns on the output. Next we need to join the ID to get company names from out list dataset. Grab another Join tool and place it after the first one. Connect the J output anchor of the first join to the Left input anchor of the 2nd join tool. Next, connect its right anchor to the output of the record ID tool of the List data stream. For its join field, set ListID2 on the left, and ListID on the right. Finally, uncheck the following to exclude them from the output; Source field from both left and right datastream, and ListID from the Right. Once all configurations are complete, run the workflow.

As we can see from the output, The company “Mydo” was matched to 3 records. The first one has a match score of 93 percent since it was all in uppercase and only has 1 extra letter. The other 2 matches were 86% and 89% because they have the Limited Acronym as suffix. You can easily check which records are matched since we’ve joined them back to get the company name. But if you wanted to create a group for each match, for example, create a group for records 1 up to 3, you can utilize the Make Group Tool.

**Make Group Tool is used to take data relationships and assemble the data into groups based on those relationships. For example, if A is matched to B, and B is matched to C, the group members for the first group will be A, B and C. Other members that do not relate on any way to the group will not be included.**

Drag a make group tool from the Join palette, and connect it to the J output anchor of the 2nd join tool. This tool has 2 simple configurations. You only need to map the ID fields for each Key. In this workflow, set ListID as the 1st Key and ListID2 as the 2nd Key. Once done, run the workflow. From the output, Group 1 contains these members, 1, 99, 100, and 101. If you go back to the output anchor of the 2nd join, we can see that those IDs pertain to the matches for the company Mydo. You can utilize this tool to create clusters, and group connected fields in your dataset.