Data Analysis and Integration

Lab 5: Approximate duplicate detection

Input data

1. Download the file **customers.csv**.

2. Open the file in a text editor to inspect its format.

3. Open the same file in **LibreOffice Calc**.

4. Use **Comma** as separator, and double quotes (") as **String delimiter**.

5. Take a moment to check if you can spot some duplicate records.

Reading the input data

6. Open a new terminal and navigate to the folder: **~/Pentaho/data-integration**

7. Start Pentaho Data Integration (PDI) with: **./spoon.sh**

8. In the **File** menu, select **New > Transformation**.

9. In the **Design** tab, expand **Input**, and drag a **CSV file input** step to the canvas.

10. Double-click the **CSV file input** to configure it:
    - In **Filename**, write **/home/aid/Downloads/customers.csv** (if you are on the VM).
    - In **Delimiter**, use a comma (**,**)
    - In **Enclosure**, use a double-quote (**"**)
    - Uncheck the **Lazy conversion** option.
    - Make sure that the **Header row present** option is checked.
    - In **File encoding**, choose **UTF-8**.

    Do not close the configuration dialog just yet.

11. Press the **Get Fields** button. This will sample some lines from the CSV file.

12. In the **Sample size** window, you can leave the default size of 100. (Why?)

13. Check that all fields have been identified correctly.

14. Preview the **CSV file input** step and check that the file contents have been read correctly.

Renaming the columns

15. In the **Design** tab, expand **Transform**, and drag a **Select values** step to the canvas.

16. Connect the **CSV file input** step to the **Select values** step (choose **Main output of step**).

17. Configure the **Select values** step as follows:
    - In the **Select & Alter** tab, press the **Get fields to select** button.
    - Use the second column (**Rename to**) to rename every field as follows:

customer_id	**customer_id_1**
first_name	**first_name_1**
last_name	**last_name_1**
address	**address_1**
city	**city_1**
country	**country_1**

18. Preview the **Select values** step and verify that the columns are being renamed correctly.

19. In the **Design** tab, expand **Transform**, and drag another **Select values** step to the canvas.

20. Connect the **CSV file input** step to the **Select values 2** step (choose **Main output of step**).

21. A **Warning** dialog will ask if you would like to distribute or copy the rows from the **CSV file input** to both destinations. Choose **Copy**.

22. Configure the **Select values 2** step as follows:
    - In the **Select & Alter** tab, press the **Get fields to select** button.
    - Use the second column (**Rename to**) to rename every field as follows:

customer_id	**customer_id_2**
first_name	**first_name_2**
last_name	**last_name_2**
address	**address_2**
city	**city_2**
country	**country_2**

23. Preview the **Select values 2** step and verify that the columns are being renamed correctly.

Comparing the records

24. In the **Design** tab, expand **Joins**, and drag a **Join Rows (cartesian product)** step to the canvas.

25. Double-click the **Join Rows (cartesian product)** and change the **Step name** to simply **Join Rows**.

26. Connect **Select values** to **Join Rows** (choose **Main output of step**).

27. Connect **Select values 2** to **Join Rows** (choose **Main output of step**).

28. Preview the **Joins Rows** step and verify that it is generating all possible pairs of records.
    *Note: When previewing, you may want to increase the **Number of rows to retrieve** to 10000.*

29. It is not necessary to compare each record with itself. Therefore, configure the **Join Rows** step with the following condition: **customer_id_1 <> customer_id_2**

30. Preview the **Joins Rows** step and verify that it is not generating pairs with the same record anymore.

31. It is not necessary to compare two records twice (i.e. both and). Therefore, configure the **Join Rows** step with the following condition: **customer_id_1 < customer_id_2**

32. Preview the **Joins Rows** step and verify that it is working according to the condition above.

Calculating the similarity measures

33. In the **Design** tab, expand **Transform**, and drag a **Calculator** step to the canvas.

34. Connect the **Join rows** step to the **Calculator** step.

35. Configure the **Calculator** step as follows:
    - Add a new field called **sim1** that calculates the **Levenshtein distance** between **first_name_1** (Field A) and **first_name_2** (Field B).
    - Add a new field called **sim2** that calculates the **Levenshtein distance** between **last_name_1** (Field A) and **last_name_2** (Field B).
    - Add a new field called **sim3** that calculates the **Levenshtein distance** between **address_1** (Field A) and **address_2** (Field B).
    - Set the **Value type** for the three fields as **Number** (i.e. real, not integer).

36. Preview the **Calculator** step and check that it is calculating the three measures correctly.

37. In the **Design** tab, expand **Scripting**, and drag a **Formula** step to the canvas.

38. Connect the **Calculator** step to the **Formula** step.

39. Double-click the **Formula** step and configure it as follows:
    - Add a new field called **sim_total** with the formula: **0.3*[sim1]+0.3*[sim2]+0.4*[sim3]**
    - Set the **Value Type** for this new field to **Number**.

40. Preview the **Formula** step and check that it is calculating the **sim_total** measure correctly.

Applying a threshold

41. Preview the **Formula** step again, but this time set the **Number of rows to retrieve** to 10000 in order to see all rows.

42. Click on top of the **sim_total** column to sort the rows by the values in that column.

43. Identify a value of **sim_total** that is a good threshold to separate duplicate from non-duplicate records. Take note of this value. Tudo o que é maior ou igual a 3.9 (sim_total) já não tem duplicados

44. In the **Design** tab, expand **Flow**, and drag a **Filter rows** step to the canvas.

45. Connect the **Formula** step to the **Filter rows** step.

46. Configure the **Filter rows** step as follows:
    - The condition is that **sim_total** must be less than the threshold value that you have previously identified.
    - Make sure that the value **Type** is set to **Number**.

47. Preview the **Filter rows** step and verify that it keeps only the duplicate records.

48. In the **Design** tab, expand **Transform**, and drag a **Select values** step to the canvas.

49. Connect the **Filter rows** step to the **Select values 3** step (choose **Result is TRUE**).

50. Configure the **Select values 3** step as follows:
    - In the **Select & Alter** tab, select the fields **customer_id_1** and **customer_id_2** only.

51. Preview the **Select values 3** step and verify that it gives the pairs of customer ids from the duplicate records.

Removing the approximate duplicates

52. In the **Design** tab, expand **Lookup**, and drag a **Stream lookup** step to the canvas.

53. Connect the **CSV file input** step to the **Stream lookup** step (choose **Main output of step**).

54. Connect also the **Select values 3** step to the **Stream lookup** step (choose **Main output of step**).

55. Configure the **Stream lookup** step as follows:
    - In **Lookup step** choose **Select values 3**
    - In **The key(s) to look up the value(s)**:
        - add the field **customer_id** with the lookup field **customer_id_2**
    - In **Specify the fields to retrieve:**
        - add the field **customer_id_1** with new name **duplicate** and type **Integer**

56. Preview the **Stream lookup** step and verify that the duplicate records have been correctly identified in the **duplicate** column.

57. In the **Design** tab, expand **Flow**, and drag a **Filter rows** step to the canvas.

58. Connect the **Stream lookup** step to the **Filter rows 2** step.

59. Configure the **Filter rows** step as follows:
    - The condition is that **duplicate IS NULL**.

60. Preview the **Filter rows 2** step and verify that it gives only non-duplicate records.

Saving the results

61. In the **Design** tab, expand **Output**, and drag a **Text file output** step to the canvas.

62. Connect the **Filter rows 2** step to the **Text file output** step (choose **Result is TRUE**).

63. Configure the **Text file output** step as follows as described in the next steps.

64. In the **File** tab:
    - In **Filename**, write **/home/aid/Downloads/customers2** (if you are on the VM)
    - Change the **Extension** from **txt** to **csv**
    - Press the button **Show filenames** to check the full path to the file that will be created.

65. In the **Content** tab:
    - Set the **Separator** to a comma (**,**)

- Set the **Enclosure** to a double-quote (**"**)
- Make sure that the option **Header** is checked.
- In **Encoding**, select **UTF-8**.

66. In the **Fields** tab:
    - Press the **Get Fields** button.
    - Then press the **Minimal width** button.
    - Delete the line with the **duplicate** field.

67. Close the **Text file output** configuration dialog with **OK**.

Saving and running the transformation

68. Save the transformation as **lab05.ktr**

69. Run the transformation by pressing the **Run** button in the toolbar.

70. In the **Execution Results** pane, switch to the **Step Metrics** tab.

71. In the **Input** column, check how many rows have entered the transformation.

72. In the **Output** column, check how many rows have exited the transformation.

73. In the **Read** and **Written** columns, check how many rows have entered and exited each step.
    - How many comparisons have been made?
    - How many pairs of duplicates have been found?
    - How many duplicates have been removed from the input data?

74. Go to the folder where the transformation has saved the **customers2.csv** file.

75. Open the file in a text editor to inspect its format.

76. Using a terminal, navigate to the same folder and execute the following command:
    diff customers.csv customers2.csv

77. You should get a list of lines that are present in the input data, but are no longer present in the output data from the transformation.

Exercise

78. Change the **Calculator** step to use the Jaro measure, and preview the results of the **Formula** step.
    *Note: Set the **Number of rows to retrieve** to 10000 in order to see all rows.*

79. Sort the rows by **sim_total** (ascending or descending?) and try to find a threshold to separate duplicate from non-duplicate records.

80. Use that threshold in the **Filter rows** step to catch the duplicate records (**sim_total >** …). 0.7108058608

81. Run the transformation again and verify that you get the same results as before.