Oracle Financial Services Financial Crime and Complaince Studio

Matching Guide

Release 8.0.8.3.0

May 2021

E91246-01





Financial Crime Graph Model Matching Guide

Copyright © 2021 Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be errorfree. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

For information on third party licenses, click here.

Document Control

| Version Number | Revision Date | Change Log |
|----------------|------------------------------|--|
| 8.0.8.2.0 | Updated: Janu- ary 2021 | Updated the <u>Scoring Method</u> section in the document. |
| 8.0.8.0.0 | Updated: Sep- tember 2020 | Updated the document for the Financial Crime Graph Model Matching Guide for v8.0.8.0.0 Release. |
| 8.0.7.4.0 | Updated: April 2020 | Updated the document for the Financial Crime Graph Model Matching Guide for v8.0.7.4.0 Release. |
| 8.0.7.3.0 | Created: March 2019 | Created the first version of the Financial Crime Graph Model Matching Guide for v8.0.7.3.0 Release. |

Table of Contents

| 1 | Introduction5 | | | | | | |
|---|---------------|--------------------------|----|--|--|--|--|
| 2 | Sco | ring Method | 7 | | | | |
| | 2.1 | Default Method | .7 | | | | |
| | 2.2 | Jaro Winkler | .7 | | | | |
| | 2.3 | ML Boosted Name Matching | .8 | | | | |
| | 2.4 | Reverse Jaro | .8 | | | | |
| | 2.5 | Fuzzy Tokenized | .9 | | | | |
| 3 | Mat | tching Rulesets | 11 | | | | |
| | 3.1 | Example | 11 | | | | |
| | 3.1.1 | Calculation of Score | 15 | | | | |

1 Introduction

In FCC Studio, data is obtained from FCDM (Financial Crime Data Model) to generate Financial Crime Graph Model. The graph model includes nodes for entities such as Customers, Accounts, Events, and Derived Entities, and edges for transactions and relationships.

Entity Resolution compares nodes with the objective to identify pairs or groups of nodes that refer to the same entity. Entity Resolution creates Similarity Edges between nodes by comparing the attributes of the nodes and identifying where the similarity is significant enough to create an edge so the nodes are linked with the graph model and can be analyzed as a single entity.

Entity matching rules are used to compare nodes of different types. For example, deduplicating customers, resolving derived entities, or linking customers or derived entities to external data such as panama papers or sanctions lists with different rules and thresholds.

For example:

A customer holds three different accounts in a bank with three different customer details.

Customer 1Customer 2Customer 3Mary MathewMary Maria
MathewMary Steve

The following table provides the customer details of Customer 1, Customer 2, and Customer 3 in a bank.

| Table | 1: | Customer | Details |
|-----------|----|--------------|---------|
| 1 4 5 1 5 | | • 40 00 1101 | Dotano |

Figure 1: Customers in a Bank

| Customer Details | Customer 1 | Customer 2 | Customer 3 |
|------------------|---------------------------|---------------------------|-----------------------|
| Source | Source System 1 | Source System 2 | Source System 3 |
| Name | Mary Mathew | Mary Maria Mathew | Mary Steve |
| Email | Mary.Mathew@gmail.c om | Mary.Mathew@gmail.c om | Mary.Steve@gmail.co m |
| Phone | Phone Number 1 | Phone Number 2 | Phone Number 3 |
| Country | United States | United States | United States |
| State | California | California | Washington |
| Address | Redwood City | Redwood City | 15th St NW |
| DOB | 1 Jan 1995 | 1 Jan 1995 | 1 Jan 1995 |
| Tax ID | Tax ID 1 | - | Tax ID 1 |

The customer details include source from which the customer data is obtained, name, Email, phone number, country, state, address, date of birth, and tax ID of the customer.

Using Entity Resolution, you can execute the Customer to Customer Ruleset on the customer data to compare the nodes such as Customer Name, Email, Phone, Country, State, Address, DOB, and Tax ID.

The result obtains an exact match on the DOB and TaxID, and fuzzy match on the Customer Name, Email ID, Address, State, and Country, and no match for the phone number.

This helps to derive to a conclusion to draw Similarity Edges between the three customers in the Bank.



Figure 2: Derived Similarity Edges for Customers in a Bank

2 Scoring Method

The scoring methods used in the entity resolution component are as follows:

- Default Method
- Jaro Winkler
- ML Boosted Name Matching
- Reverse Jaro
- Fuzzy Tokenized

2.1 Default Method

The distance is computed by finding the number of edits which transforms one string to another. The transformations allowed are as follows:

Insertion: Adding a new character

Deletion: Deleting a character

Substitution: Replace one character with another

By performing these operations, the algorithm attempts to modify the first string to match the second one. The final result obtained is the edit distance.

For example:

- textdistance.levenshtein('arrow', 'arow') 1
- textdistance.levenshtein.normalized similarity('arrow', 'arow')0.8

Here, if you insert single 'r' in string 2, that is, 'arow', it becomes same as the string 1. Hence, the edit distance is 1. Similar with Hamming distance, you can generate a bounded similarity score between 0 and 1. The similarity score obtained is 80%.

2.2 Jaro Winkler

This algorithms gives high scores for the following strings:

The strings that contain same characters, but within a certain distance from one another.

The order of the matching characters is same.

To be precise, the distance of finding similar character is one character less than half of the length of the longest string. So if the longest string has a length of five, a character at the start of the string 1 must be found before or on $((5/2)-1) \sim 2nd$ position in the string 2. This is considered a valid match. Hence, the algorithm is directional and gives high score if matching is from the beginning of the strings.

For example:

- textdistance.jaro_winkler("mes", "messi") 0.86
- textdistance.jaro_winkler("crate", "crat") 0.96
- textdistance.jaro_winkler("crate", "atcr") 0.0

In first case, as the strings are matching from the beginning, high score is given. Similarly, in the second case, only one character was missing and that too at the end of the string 2, hence a very high score is given. In third case, the last two character of string 2 are rearranged by bringing them at front and hence results in 0% similarity.

2.3 ML Boosted Name Matching

This scoring method uses Machine Learning to calculate a similarity score between two candidate name strings.

It uses the Catboost model, which is an ensemble of decision trees built using gradient boosting embedding algorithm. The input to the model are numerical features. The features are calculated as various similarity metrics between the two candidate names.

For example, Q-gram distance, longest common subsequence, Levenshtein distance, Jaro-Winkler distance, set letter distance, Editex, and so on. The output of the model is a similarity score between 0 and 1. For this, a pre-trained model is provided. The pre-trained model is built from names from publicly available datasets depending on whether we mention the names of the datasets or not, we should use the term "publicly available" accordingly. The names are transformed, by inserting typos, concatenations, abbreviations, and so on.

The the original and transformed names are matched by a smart filter to produce a list of possible matches called candidates, along with information indicating whether two names started from the same name (ground truth). Then you can calculate the numerical similarity feature metrics of the candidate names.

The name strings are dropped, and the numerical features along with ground truth is used for training of the ensemble of decision trees model (Catboost model).

Following are the examples using the pre-trained model:

- Name 1: Carolyn Joyce
- Name 2: Caroline Joyce
- Resulting score: 0.9821176658138586

Other example,

- Name 1: Donatella Collombini
- Name 2: Donatello di Betto Bardi
- Resulting score: 0.0000005237652440222384

2.4 Reverse Jaro

This algorithm tokenises source and target tokens, then uses Jaro Winkler algorithm to calculate the score between tokens, and then consolidate the scores to a single score.

Tokenization:

The source and target strings are tokenised with each space. For instance, if the source name is **ANTONY EDWARD STARK** and target name is **TONY STARK**, we tokenise source name to source array as, **EDWARD**, **STARK** and target string as **TONY**, **STARK**.

Scoring:

The array is represented as a 2-d matrix, with source token array as rows and target token array as columns and then scores between those tokens will be calculated using Jaro Winkler algorithm.

Table 1: Customer Details

| Customer Details | ΤΟΝΥ | STARK |
|------------------|------|-------|
| ANTONY | 94 | 14 |
| EDWARD | 43 | 32 |
| STARK | 18 | 100 |

The Maximum weighted row scores and maximum column scores will be calculated For the example in the table,

MaximumWeightedRowScore are as follows:

- 94 (maximum of 94, 14)
- 43 (maximum of 43, 32)
- 100 (maximum of 18,100)

MaximumWeightedColScore are as follows

- 94 (maximum of 94, 43, 18)
- 100 (maximum of 14,32,100)

To get the final score, you can calculate the sum of maximumWeightedRowScore and MaximumWeightedColumnScore (237+194) divided by the sum of row size and column size (3+2). In this case 431/5 = 86.2

2.5 Fuzzy Tokenized

This algorithm tokenises source and target tokens, then uses Jaro Winkler algorithm to calculate the score between tokens, and then consolidate the scores to a single score.

Tokenisation:

The source and target strings are tokenised with each space. For instance, if the source name is **ANTONY EDWARD STARK** and target name is **TONY STARK**, we tokenise source name to source array as **ANTONY**, **EDWARD**, **STARK** and target string as **TONY**, **STARK**.

Always the string with maximum number of tokens will be passed as source token array and the other will be target token array.

Scoring:

The array is represented as a 2-d matrix, with source token array as rows and target token array as columns and then scores between those tokens will be calculated using Jaro Winkler algorithm.

The Maximum weighted row scores and maximum column scores will be calculated

Table 2: Customer Details

| Customer Details | TONY | STARK |
|------------------|------|-------|
| ANTONY | 94 | 14 |
| EDWARD | 43 | 32 |
| STARK | 18 | 100 |

- STARK in third row matching against STARK in second column
- STARK in second column matching against STARK in third row

In this scenario both returns the same score for same source and target token, so we won't consider this value twice.

- Such instances will be counted as a resCount (residue Count).
- Similar to reverse jaro, MaximumWeightedColScore score is calculated, and Weighted score of Row without duplicates will be calculated as MaximumWeightedResRowScore

Then finally similar to Reverse Jaro, we calculate

Fuzzy tokenised score = (MaximumWeightedColScore + MaximumWeightedResRowScore) / (colSize + resCount)

only if the score is greater than the threshold.

In othercase,

Word Match Percentage is calculated. For more information, see this <u>link</u>.

Then resScore is calculated,

which is (MaximumWeightedRowScore * (1-threshold)) / (resCount +1)

After the calculations, we will count the no. of tokens in column which has score > threshold. if (count >= 2) then the **Fuzzy tokenised score = threshold + (word match percentage * resScore)**

3 Matching Rulesets

Each ruleset comprises of multiple rules. The ruleset compares the attributes that are defined in the rules for the source entity with the target entity.

The following table provides the list of rulesets that are packaged with the FCC Studio application.

| Ruleset Name | Source Node Type | Target Node Type |
|--|------------------|---------------------------|
| Customer To Customer Match | customer | customer |
| Customer To Derived Entity | customer | derived_entity |
| Derived Entity To Derived Entity | derived_entity | derived_entity |
| Customer To Ext Source - Offshore | customer | external_entity_offshore |
| Customer To Ext Source - Bahamas | customer | external_entity_bahamas |
| Customer To Ext Source - Paradise | customer | external_address_paradise |
| Customer To Ext Source - Panama | customer | external_entity_panama |
| Customer To Ext Source - Offshore Addr | customer | external_address_offshore |
| Customer To Ext Source - Bahamas Addr | customer | external_address_bahamas |
| Customer To Ext Source - Paradise Addr | customer | external_address_panama |
| Customer To Ext Source - Panama Addr | customer | external_address_paradise |

Table 1: List of Rulesets

Each ruleset contains pre-defined source and target node types. Each ruleset is used to compare the parameters/attributes of the source and target node types to obtain a match.

3.1 Example

The following table provides the customer details of Customer 1, Customer 2, and Customer 3 in a bank.

| Customer Details | Customer 1 | Customer 2 | Customer 3 |
|------------------|---------------------------|---------------------------|-----------------------|
| Source | Source System 1 | Source System 2 | Source System 3 |
| Name | Mary Mathew | Mary Maria Mathew | Mary Steve |
| Email | Mary.Mathew@gmail.c om | Mary.Mathew@gmail.c om | Mary.Steve@gmail.co m |
| Phone | Phone Number 1 | Phone Number 2 | Phone Number 3 |
| Country | United States | United States | United States |
| State | California | California | Washington |
| Address | Redwood City | Redwood City | 15th St NW |

Table 2: Customer Details

| DOB | 1 Jan 1995 | 1 Jan 1995 | 1 Jan 1995 |
|--------|------------|------------|------------|
| Tax ID | Tax ID 1 | - | Tax ID 1 |

The customer details include source from which the customer data is obtained, name, Email, phone number, country, state, address, date of birth, and tax ID of the customer.

The Customer to Customer Match ruleset compares the attributes defined for the source (customer) and target (customer) entities of each rule. If the score of the combination of the result obtained for all the rules in a ruleset is equal to or greater than the threshold set for the ruleset, a Similarity Edge is formed between the source and the target entity.

The Customer to Customer Match ruleset is given a follows:

Figure 1: Ruleset Details

| CRACLE [®] Crime & Compliance Studio Search Hidebooks | | | | | | |
|--|--------------------|----------------------------|-------------------------------------|--------------------|--------------------|------------|
| Ruleset Details | | | | | Save | Back Reset |
| Name Customer to Customer Match | | Description Description | Scoring Aggregation Type Maximum | | Set Threshold 1 | |
| Rules Create | Source Customer | | | Target Customer | | |

The following table provides the details of the Customer to Customer Ruleset.

| Ruleset | Rules | Parameters/Attributes |
|---------------------------------|------------------------|---|
| Customer to Customer Ruleset | Name and Address Match | Name, Alias, Concatenated Name Email Phone Country State Address DOB Tax ID |
| Customer to Customer Ruleset | Tax ID | Tax ID |

Table 3: Customer to Customer Ruleset

The Customer to Customer ruleset comprises of two rules, namely Name and Address Match and Tax ID. Each of these rules are applicable to pre-defined parameters/attributes. The rules are used to compare the parameters/attributes of the source and target node to obtain a match.





The following table provides the how the Name and Address Match rule works.

| Param- eters/ Attri- butes | Source Entity | Target Entity | Matc h Type | Weigh tage | Thres hold | Description |
|--|------------------|------------------|----------------|---------------|---------------|--|
| Name, Alias, Concat- enated Name | Customer | Customer | Fuzzy | 0.4 | 0.5 | The Name, Alias, and Concate- nated Name attributes of the source entity (Customer) is com- pared with the target entity (Cus- tomer) to obtain a fuzzy match. If the fuzzy match generates a result that is equal to or greater than the threshold value (0.5), a weightage of 0.4 is contributed to this match. |
| Email | Customer | Customer | Exact | 0.2 | 1 | The Email address of the source entity (Customer) is compared with the target entity (Customer) to obtain an exact match. If an exact match is obtained, a weightage of 0.2 is contributed to this match. |

Table 4: Name and Address Match Rule

Table 4: Name and Address Match Rule

| Param- | Source | Target | Matc h | Weigh | Thres | Description |
|--------|--------|--------|--------|-------|-------|-------------|
| eters/ | Entity | Entity | Туре | tage | hold | |
| Attri- | | | | | | |
| butes | | | | | | |
| | | | | | | |

| Phone | Customer | Customer | Exact | 0.1 | 1 | The phone number of the source entity (Customer) is compared with the target entity (Customer) to obtain an exact match. If an exact match is obtained, a weightage of 0.1 is contributed to this match. |
|----------|----------|----------|-------|------|-----|---|
| Country | Customer | Customer | Exact | 0.05 | 1 | The country of the source entity (Customer) is compared with the target entity (Customer) to obtain an exact match. If an exact match is obtained, a weightage of 0.05 is contributed to this match. |
| State | Customer | Customer | Exact | 0.05 | 1 | The state of the source entity (Customer) is compared with tar- get entity (Customer) to obtain an exact match. If an exact match is obtained, a weightage of 0.05 is contributed to this match. |
| Addres s | Customer | Customer | Fuzzy | 0.1 | 0.6 | The address of the source entity (Customer) is compared with the target entity (Customer) to obtain a fuzzy match. If the fuzzy match generates a result that is equal to or greater than the threshold value (0.6), a weightage of 0.1 is contributed to this match. |
| DOB | Customer | Customer | Exact | 0.1 | 1 | The date of birth of the source entity (Customer) is compared with the target entity (Customer) to obtain an exact match. If an exact match is obtained, a weightage of 0.1 is contributed to this match. |

For the Name and Address Match rule, the Source and Target entity is Customer, and the corresponding parameters/attributes are Name, Alias, Concatenated Name, Email, Phone, Country, State, Address, and Date of Birth. Based on the match type, the parameters/attributes of the source entity is compared with the target entity to obtain a match and the result contributes to the total result obtained for all the matches of the rule.

| Figure | 3: | Тах | ID | Rule |
|--------|----|-----|----|------|
|--------|----|-----|----|------|

| Tax ID | | Match On TaxID | 1 |
|--------|-------------------|-------------------|---|
| | | Exact, Weight = 1 | |
| | (Customer.Tax ID) | | |

The following table provides the how the Tax ID rule works.

| Parameters/ Attributes | Source Entity | Target Entity | Match Type | Weight age | Description |
|---------------------------|---------------|---------------|------------|---------------|--|
| Tax ID | Customer | Customer | Exact | 1 | The Tax ID of the source entity (Customer) is com- pared with the target entity (customer) to obtain an exact match. If an exact match is obtained, a weightage of 1 is contributed to this match. |

Table 5: Tax ID Rule

For the Tax ID rule, the Source and Target entity is Customer, and the corresponding parameters/ attribute is Tax ID. Based on the match type, the parameters/attributes of the source entity is compared with the target entity to obtain a match.

3.1.1 Calculation of Score

The following table provides details on how to calculate the score for the Name and Address Match rule.

| Customer Details | Customer 1 | Customer 2 | Score | Weight (From Rui | Weighted le) Score |
|---------------------|---------------------------|---------------------------|-------|---------------------|-----------------------|
| Name | Mary Mathew | Mary Maria Mathew | 93.07 | 0.4 | 37.22 |
| Email | Mary.Mathew@gma il.com | Mary.Mathew@gma il.com | 100 | 0.2 | 20 |
| Phone | Phone Number 1 | Phone Number 2 | 100 | 0.1 | 10 |
| Country | United States | United States | 100 | 0.05 | 5 |
| State | California | California | 100 | 0.05 | 5 |
| Address | Redwood City | Redwood City | 100 | 0.1 | 10 |
| DOB | 1 Jan 1995 | 1 Jan 1995 | 100 | 0.1 | 10 |
| | | | | Total=1 | Total=97.22 |

Table 6: Calculation of Score

The weightage obtained from the Name and Address Match rule contributes to the total weighted score.

The total score obtained is greater than the rule threshold of 40%, a Similarity Edge is created between Customer1 and Customer 2. Similar calculation is performed for all possible combination of customers like Customer 2 and Customer 3, Customer 1 and Customer 3.OFSAA Support

Raise a Service Request (SR) in <u>My Oracle Support (MOS)</u> for queries related to the OFSAA applications.

Send Us Your Comments

Oracle welcomes your comments and suggestions on the quality and usefulness of this publication. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most about this manual?

If you find any errors or have any other suggestions for improvement, indicate the title and part number of the documentation along with the chapter/section/page number (if available) and contact the Oracle Support.

Before sending us your comments, you might like to ensure that you have the latest version of the document wherein any of your concerns have already been addressed. You can access My Oracle Support site that has all the revised/recently released documents.

