A STRATEGIC ANALYSIS FOR THE DYNAMIC DE-DUPLICATION OF CUSTOMER DATA USING FUZZY MATCH OR SEARCH STRATEGIES

[1]A. Ghouse Mohiddin, [2]S. Ramakrishna
[1]Research Scholar, Dept. of Computer Science, Dravidian University, Kuppam, A.P., India
[2] Dept. of Computer Science, Sri Venkateswara University, Tirupathi, A.P., India.

Abstract : As digital data is growing tremendously, cloud storage services are gaining popularity since they promise to provide convenient and data storage services that can be accessed anytime, from anywhere. These huge size of data require some practical platforms for the storage, processing and availability and cloud technology over's all the potentials to full- Although data dynamic deduplication removes data redundancy and data replication by storing only a single copy of previously duplicated data [2], Data deduplication framework, with the goal of preserving to preserve the privacy of data in the cloud while ensuring that the perform data deduplication without compromising the data privacy and security. Data Analyst can be use to analyze, profile, and account data in an enterprise. We can perform column and rule profiling, score carding, bad record and duplicate record management. Reference data can include accurate and standardization values that can be used by analysts and developers in cleansing and validation rules. Standardize once the problems with the data have been identified, standardization process to cleanse, standardize, enrich and validate customer data. An identify duplicate records in Customer data using a variety of matching techniques algorithms (Fuzzy logic). An automatically or manually consolidate the matched records. [7]. Matching will identify related or duplicate records within a dataset or across two datasets. Matching scores records between 0 and 1 on the strength of the match between them, with a score of 1 indicating a perfect match between records. The Fuzzy algorithms is to provide values in selected input columns and calculates match scores representing the degrees of similarity between the pairs of values [10,11].

IndexTerms **- Data Profiling, Data Standardization, Tokenization, Math and Merge, Match Ruleset and Match Rule.**

I. INTRODUCTION

Data profiling is a specific form of data analysis customer data to detect and characterize important features of data sets. [4] . Data Analyst can be use to analyze, profile, and account data in an enterprise. We can perform column and rule profiling, score carding, bad record and duplicate record management. Reference data can include accurate and standardization values that can be used by analysts and developers in cleansing and validation rules. Standardize once the problems with the data have been identified, standardization process to cleanse, standardize, enrich and validate customer data. Identify duplicate records in Customer data using a variety of matching techniques algorithms (Fuzzy logic). An automatically or manually consolidate the matched records.

Matching will identify related or duplicate records within a dataset or across two datasets. Matching scores records between 0 and 1 on the strength of the match between them, with a score of 1 indicating a perfect match between records [12]. The Fuzzy algorithms is to provide values in selected input columns and calculates match scores representing the degrees of similarity between the pairs of values.

II. II. TRUST SCORE AND VALIDATION RULES

While configuring column properties, specify which column(s) will use trust to determine the most reliable value when different source systems provide different values for the same cell. Several source systems may contain attributes that correspond to the same column in a base object table. For example, several systems may store a customer's address.

Trust is a designation the confidence in the relative accuracy of a particular piece of data. For each column from each source, you can define a trust level represented by a number between 0 and 100, with zero being the least trustworthy and 100 being the most trustworthy. Trust takes into account the age of data, how much its reliability has decayed over time, and the validity of the data. Trust is used to determine survivorship (when two records are consolidated), and whether updates from a source system are sufficiently reliable to update the master record. A trust level is a number between 0 and 100. By itself, this number has no meaning. It has meaning only when compared with another trust number.

2.1 Data Reliability Decays Over Time

The reliability of data from a given source system can decay (diminish) over time. In order to reflect this fact in trust calculations, Master Data Hub allows to configure decay characteristics for trust-enabled columns. The decay period is the amount of time that it takes for the trust level to decay from the maximum trust level to the minimum trust level.

2.2 Trust Calculations

The load process calculates trust for trust-enabled columns in the base object. For records with trust-enabled columns, the load process assigns a trust score to cell data. This trust score is initially based on the configured trust settings for that column. The trust score may be subsequently downgraded when the load process applies validation rules—if configured for a trust-enabled column—after the trust calculations.

2.3 Overriding Trust Scores

Data stewards can manually override a calculated trust setting if they have direct knowledge that a particular values correct. Data stewards can also enter a value directly into a record in a base object.

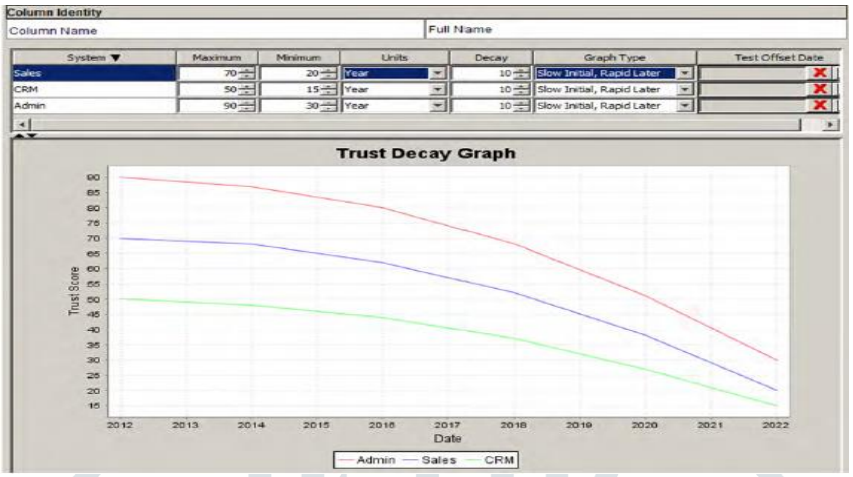
2.4. Trust Properties

Maximum Trust: The maximum trust (starting trust) is the trust level that a data value will have if it has just been changed. By setting the maximum trust level relatively high, we can ensure that changes in the source systems will usually be applied to the base object.

Minimum Trust: The minimum trust is the trust level that a data value will have when it is old (after the decay period has elapsed). This value must be less than or equal to the maximum trust.

Units: Specifies the units used in calculating the decay period—day, week, month, quarter, or year.

Decay: Specifies the number (of days, weeks, months, quarters, or years) used in calculating the decay period. For Example, the best graph view, limit the decay period you specify to between 1 and 100.



2.4 Figure- Configure Trust Score

2.5. Configuring Validation Rules

A validation rule downgrades trust for a cell value when the cell value matches a given condition.

Each validation rule specifies:

A condition that determines whether the cell value is valid an action to take if the condition is met (downgrade trust by a certain percentage)

For example, the following validation rule:

Downgrade trust on First Name IS NULL by 20% or if Length < 3'

Downgrade trust on First Name by 20%

If the Reserve Minimum Trust flag is set for the column, then the trust cannot be downgraded below the column'sminimum trust. We use the Schema Manager to configure validation rules for a base object. Validation rules are executed during the load process, after trust has been calculated for trust-enabled columns in the base object. If validation rules have been defined, then the load process applies them to determine the final trust scores, and then uses the final trust values to determine whether to update records in the base object with cell data from the updated records.

Validation rules downgrade trust scores according to the following algorithm:

- Final trust = Trust - (Trust * Validation Downgrade / 100)
- For example, with a validation downgrade percentage of 20%, and a trust level calculated at 60.
- Final Trust Score = 60 - (60 * 20 / 100)
- The final trust score is:
- Final Trust Score = 60 - 12 =48.

III. IMPLEMENTATION

3.1. MATCH PROCESS/TOKENIZATION PROCESS

The Tokenization process runs only on Base Objects with Fuzzy match strategy. Tokenization is the process of identifying the Match pairs between the records (Fuzzy Match Key) by generating and comparing tokens. Tokenization generates 1-20 tokens for each record of Fuzzy Match Key depending on the Key width and the length of the record. A token is an 8 char system generated value that are stored in BaseObject_STRP table(column-SSA_KEY).If there is a match between the tokens of two records, then they are identified as Match Pairs. The process of matching is only executed on these match pairs. So tokenization is basically the first step in Matching Process before executing the match process. Dirty table contain only one columns i.e. Rowid_object and Strip table contain 3 columns i.e.

SSA_Key : It generate tokens for encrypted format

SSA_DATA: It generate input data from all columns for match column configuration.

PREFRRED_KEY_IND: It indicate which valid data, 1- Matched record, 0- Not-Matched records.

Tokens are generated based on ***"Fuzzy Match Key"*** **(Person_Name, Org_Name, and Address_ Part1)** what we have defined in Match & Merge and Bucketing process will take place based on the Match columns what we have defined and it will populate the _STRP as SSA_KEY and SSA_DATA.

3.2 Match/Search Strategy

It Specifies the reliability of the match versus the performance require, Fuzzy/ Exact. An Exact match / search strategy is faster, but an exact match will miss some matches if the data is imperfect. An Exact Match Columns may be used when the Match/Search Strategy for the base object is set to Exact or Fuzzy. In this case, the Sub Category base object uses Exact as the strategy. However, it is not possible to use a Fuzzy Match Column with an Exact Match/Search Strategy.

Configuring Match Path and Match Columns for related records

The customer base object on its own does not really contain enough data to identify realistic matches- if we try to identify matches using only person/organization names, we will overmatch i.e. we will end up treating records for different and distinct customer as all belonging to one customer.To improve the match quality, we need to provide the match engine with as much information about the customer as possible. We will be including data from the Address base object by defining a match path.

Allows to traverse the hierarchy between records—whether that hierarchy exists between base objects (inter-table paths) or within a single baseobject (intra-table paths). Match paths are used for configuring match column rules involving related records in either separate tables or in the same table.

The Match Path Component as Root (Sub_Category). This is because sub-categories will only be matched using data from the Sub_Category table; i.e. we don't want the match to use data from any related child tables.

- Configuring Match Path
- Relationship Base Objects
- Check for missing children
- Inter-Table Paths

3.3. Match Columns –

A column that is used in a match rule for comparison purposes. Each match column is based on one or more columns from the base object.

3.3.1. Key Type

This is the main criterion for the search that builds the initial list of potential match candidates. This key type should be based on the main type of data that is in physical column(s) that make up the fuzzy match key.

For a fuzzy-match base object, you can select one of the following key types:

Person_Name Used if your fuzzy match key contains data for individuals only.

Organization_Name Used if your fuzzy match key contains data for organizations only, [[[;=-[;or if it contains data for both organizations and individuals.

Address_Part1 Used if your fuzzy match key contains address data to be consolidated.

3.3.2. Key Width:

The match key width determines the thoroughness of the analysis of the fuzzy match key, the number of possible match candidates returned, and how much disk space the keys consume. Key widths apply to fuzzy-match objects only.

Standard Appropriate for most fuzzy match keys, balancing reliability and space usage.

Extended Might result in more match candidates, but at the cost of longer processing time to generate keys. This option provides some additional matching capability due to the concatenation of columns.

Limited Trades some match reliability for disk space savings. This option might result in fewer match candidates, but searches can be faster. This option works well if you are willing to under match for faster searches that use less disk space for the keys. Limited keys match fewer records with word-order variations than standard keys. This choice provides a subset of the Standard key set, but might be the best option if disk space is restricted or the data volume is extremely large. Preferred Generates a single key per base object record. This option trades some match reliability for performance(reduces the number of matches that need to be performed) and disk space savings (reduces the size of the match key table). Depending on characteristics of the data, a preferred key width might result in fewer match candidates.

3.4. Match/Search Rule set:

A match rule set is a logical collection of match column rules that have some properties in common. Match rule sets are associated with match column rules only—not primary key match rules. The match process uses only one match rule set per execution.

Match rule sets allow to accommodate different match column rule requirements at different times. For example, We might use one match rule set for an initial data load and a different match rule set for subsequent incremental loads.

Example issues include a match rule set that:

- Is identical to an already existing match rule set
- Is empty—no match column rules have been added
- It contains no fuzzy-match column rules for a fuzzy-match base object
- It contains one or more fuzzy-match columns but no exact-match column (can impact match performance)
- It contains fuzzy and exact-match columns with the same source columns

3.4.1. Match Rule Set Properties

Name: The name of the rule set. Specify a unique, descriptive name.

Search Levels: Used with fuzzy-match base objects only. When we configure a match rule set, we define a search level that instructs MDM Hub on how stringently and thoroughly to search for candidate matches .The goal of the match process is to find the optimal number of matches for customer data. Under matching which misses relevant matches, or Overmatching which generates too many matches, including matches that are not relevant. For any Name or Address in a fuzzy match key, MDM Hub uses the defined search level to generate different key ranges for the purpose of determining which records are possible match candidates—and to which records the match column rules will be applied.

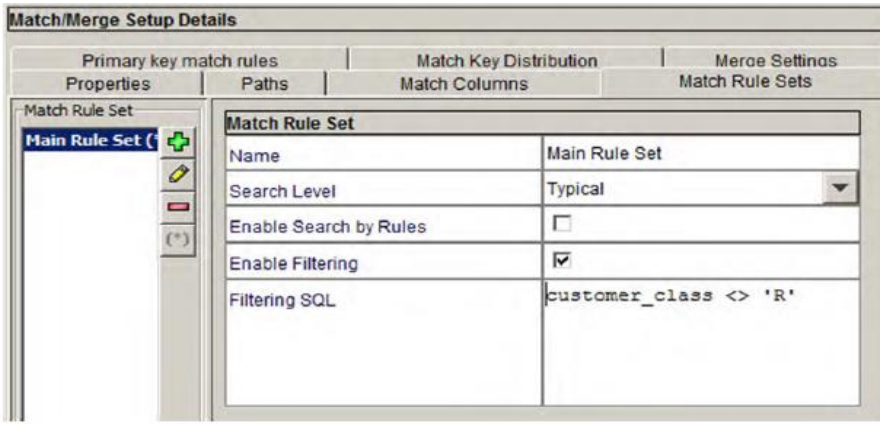
There 4 Types of Search Level

Narrow: Most stringent level in searching for possible match data. This search level is fast, but it can result in fewer matches than other search levels might and possible result in under matching. Narrow can be appropriate if our data set is relatively correct and complete, or for very large data sets with highly match data.

Typical: Appropriate for most rule sets.

Exhaustive: Generate a larger set of possible match candidates then the typical level. This can result in more matches than other search levels might generate, possible result in overmatching and take more time. This level might be appropriate for smaller data sets that are less complete.

Extreme: This level might be appropriate for smaller data sets that are less complete or to identity the highest possible number of matching records.



3.4. Figure - Match Ruleset.

3.5. Match Rules:

A fuzzy match rule is required for data that might have inconsistencies. If the quality of the data is good, we can configure exact match rules. If we want to run a large batch job and at the same time ensure optimal performance, we might want to configure filtered match rules. Filtered match rules are exact match rules that use the fuzzy match key in addition to exact match columns. We can define a match rule for exact or fuzzy matching. An exact match rule matches records that have identical values in match columns. A fuzzy match rule matches similar records based on probabilistic match determinations that consider likely variations in data patterns, such as misspellings, transpositions, omissions, and phonetic variations. The match rules are configure depend on the characteristics of the data, and the particular match and merge requirements.

Match Rules

Ru...	Auto	Type	Accep...	Purpose(Level)	Columns
1	No	Fuzzy	0	Organization(Loose)	Address_Part1 (Fuzzy) Address_Part2 (Fuzzy) Customer_Class {'O','R'} Organization_Name (Fuzzy)
2	No	Fuzzy	0	Resident(Loose)	Address_Part1 (Fuzzy) Address_Part2 (Fuzzy)(+2) Customer_Class {'I'} Generation (∅↔∅) Person_Name (Fuzzy)
3	No	Fuzzy	0	Person_Name(Loose)	Customer_Class {'I'} Generation (∅↔∅) Person_Name (Fuzzy) State

Figure 3.5 - Configure Match rules

IV. CONCLUSION

Data deduplication is a process of identifying the redundancy in data and then removing customer data. A set of processes that measure and improve the quality of important data on an ongoing basis, ensures that data dependent business processes and applications deliver expected results. Data Standardization is the problems with the data have been identified, to cleanse the data through standardization process, enrichment and validate the good data. Matching will identify related or duplicate records within a dataset or across two datasets. Matching scores records between 0 and 1 on the strength of the match between them, with a score of 1 indicating a perfect match between records. The Fuzzy algorithms is to provide values in selected input columns and calculates match scores representing the degrees of similarity between the pairs of values.

REFERENCES

[1] Chaudhuri, S., Dayal, U.: An Overview of Data Warehousing and OLAP Technology. ACM SIGMOD Record 26(1), 1997.
[2] Batini, C.; Lenzerini, M.; Navathe, S.B.: A Comparative Analysis of Methodologies for Database Schema Integration.In Computing Surveys 18(4):323-364, 1986.
[3] Savasere, A.; Omiecinski, E.; Navathe, S.: An Efficient Algorithm for Mining Association Rules in Large Databases
[4] Christen P. Febrl: an open source data cleaning, deduplication and record linkage system with a graphical user interface. Las Vegas: ACM SIGKDD; 2008. p. 1065–8
[5] A.Ghouse Mohiddin, S.Ramakrishna "Tactics for Dynamic Data Cleansing and Data Profiling Using Dimensions for Data Quality Assessment" (IJCSE) International Journal on Computer Science and Engineering" Volume-6, Issue-4 E-ISSN: 2347-2693.
[6] A.Ghouse Mohiddin, S.Ramakrishna, Sheik Mohamed "Probabilistic Latent Semantic Data Analysis for Grouping and Matching Process using Field Matching Algorithm" IJRECE VOL. 6 ISSUE 2 APR.-JUNE 2018 ISSN: 2393-9028 (PRINT) | ISSN: 2348-2281 (ONLINE)
[7] Monge, A. E.; Elkan, P.C.: The Field Matching Problem: Algorithms and Applications. Proc. 2nd Intl. Conf. Knowledge Discovery and Data Mining (KDD), 1996.
[8] Hernandez MA, Stolfo SJ. The Match amd Merge problem for large databases. San Jose: ACM SIGMOD; 1995. p. 127–38.
[9] Hernandez, M.A.; Stolfo, S.J.: Real-World Data is Dirty: Data Cleansing and the Merge/Purge Problem. Data Mining and Knowledge discovery 2(1):9-37, 1998.
[10] A.Ghouse Mohiddin, S.Ramakrishna "Tactics for Dynamic De-Duplication of Customer Data an applies Match Tuning Techniques using Fuzzy Algorithm" IJRECE VOL. 6 ISSUE 2 APR.-JUNE 2018 ISSN: 2393-9028 (PRINT) | ISSN: 2348-2281 (ONLINE).