DSE 203 (Fall 2020)

Value Matching as a Machine Learning Problem – A Case Study

Value-Matching as an ML Problem

- Value matching can be viewed as a classification problem
  - In value matching, given a set of data item x_1, x_2, x_3 … and a set of candidates Y, the system returns all pairs (x_i, y_j) such that y_j is "close to" x_i
  - In other words, for each possible (x_i, y_j) pair, the pair either is a match, or it is not
  - A suitable set of features (descriptors of x_i, y_j) and a model, the value matcher can be thought to behave as a binary classifier that assigns true or false to a candidate string
- Three decisions
  - The Features
  - The Learning Model
  - The Fitness Criterion

Training Scheme

A Use Case – a DB Cleaning Exercise

- Feature Engineering

a. Phonetic Similarity
b. Textual Similarity (typos, etc.)
c. Nickname (Mike/Michael)
d. Missing spaces/hyphens (Mary Ellen/MaryEllen/Mary-Ellen)
e. Initials (JE Smith/James Earl Smith)
f. Out of order components (Diaz Carlos Alfonzo/Carlos Alfonzo Diaz)
g. Name Swap (Jesse Jones/Jones Jesse)
h. Different Name Split (First: Dick, Last: Van Dyke/First: Dick Van, Last: Dyke)
i. Truncated Name (Charles Livingston/Charles Living)
j. Missing Name (John Albert Lewis/John Lewis)
k. Maiden name addition (or any other additional last name)

We … matching between two identifiers … For a substantial part of the DB, we hold gender data and age data, but very often this info is missing. So I am basically left with just the actual names. So how can I be sure that two names belong to the same person?

- For Scalability
  - Only match people within the same zip code

https://towardsdatascience.com/hybrid-fuzzy-name-matching-52a4ec8b749c

Distance Measures To Features

- Nicknames
  - Create a dictionary of names and nicknames. A binary feature, 1 if one of the person's name is the nickname of the other person's name, and 0 otherwise.
- Textual Similarity
  - Jaro-Winkler Distance, Hamming Distance, Damerau-Levenshtein Distance and Levenshtein Distance
- Phonetic Similarity
  - NYSIIS and Double Metaphone measures
- Out_of_order_components_score
  - full_name1_splits = re.split("\W+|_", full_name1)
  - full_name2_splits = re.split("\W+|_", full_name2)
  - len(Counter(full_name2_splits) & **Counter**(full_name1_splits)) / float(max(len(full_name1_splits), len(full_name2_splits)))
- …

C1: Counter({'b': 3, 'a': 2, 'c': 1})
C2: Counter({'a': 2, 'b': 1, 'e': 1, 'h': 1, 'l': 1, 'p': 1, 't': 1})
C1 & C2 → Intersection (taking positive minimums):
Counter({'a': 2, 'b': 1})

New York State Identification and Intelligence System

- Transformation Phase
  1. If the first letters of the name are
     - 'MAC' then change these letters to 'MCC'
     - 'KN' then change these letters to 'NN'
     - 'K' then change this letter to 'C'
     - 'PH' then change these letters to 'FF'
     - 'PF' then change these letters to 'FF'
     - 'SCH' then change these letters to 'SSS'
  2. If the last letters of the name are
     - 'EE' then change these letters to 'Y <blank>'
     - 'IE' then change these letters to 'Y <blank>'
     - 'DT' or 'RT' or 'RD' or 'NT' or 'ND' then change these letters to 'D <blank>'
  3. The first character of the NYSIIS code is the first character of the name.

New York State Identification and Intelligence System

4. Set the pointer to point to the second character of the name. Considering the position of the pointer, only one of the following statements can be executed.

Step 5 ➡

   1. If blank then go to rule 7.
   2. If the current position is a vowel (AEIOU) then if equal to 'EV' then change to 'AF' otherwise change current position to 'A'.
   3. If the current position is the letter
      'Q' then change the letter to 'G'
      'Z' then change the letter to 'S'
      'M' then change the letter to 'N'
   4. If the current position is the letter 'K' then if the next letter is 'N' then replace the current position by 'N' otherwise replace the current position by 'C'
   5. If the current position points to the letter string
      'SCH' then replace the string with 'SSS'
      'PH' then replace the string with 'FF'
   6. If the current position is the letter 'H' and either the preceding or following letter is not a vowel (AEIOU) then replace the current position with the preceding letter.
   7. If the current position is the letter 'W' and the preceding letter is a vowel then replace the current position with the preceding position.
   8. If none of these rules applies, then retain the current position letter value.
6. If the current position letter is equal to the last letter placed in the code then set the pointer to point to the next letter and go to step 5. The next character of the NYSIIS code is the current position letter. Increment the pointer to point at the next letter. Go to step 5.
7. If the last character of the NYSIIS code is the letter 'S' then remove it.
8. If the last two characters of the NYSIIS code are the letters 'AY' then replace them with the single character 'Y'.
9. If the last character of the NYSIIS code is the letter 'A' then remove this letter.

Double Metaphone (Phonetic Encoding)

string1	dblmeta_s1	string2	compare
My String	["MSTRNK","MSTRNK"]	my string	TRUE
judge	["JJ","AJ"]	juge	TRUE
knock	["NK","NK"]	nock	TRUE
white	["AT","AT"]	wite	TRUE
record	["RKRT","RKRT"]	record	TRUE
pair	["PR","PR"]	pear	TRUE
bookkeeper	["PKPR","PKPR"]	book keeper	FALSE
test1	["TST","TST"]	test123	TRUE
the end.	["0NT","TNT"]	the endâ€¦.	TRUE
a elephant	["ALFNT","ALFNT"]	an elephant	FALSE

https://pypi.org/project/Fuzzy/

It uses C Extensions (via Cython) for speed.

The algorithms are:

- Soundex
- NYSIIS
- Double Metaphone Based on Maurice Aubrey's C code from his perl implementation.

Learning Model and Scoring

- Ensemble Methods
  - Random Forest, Gradient Boosting and XGBoost
    - Why three models?
  - Hyperparameter Optimization, using sklearn's GridSearchCV
- Scoring
  - Since this is a binary classification problem use ***precision*** as the score
    - Worse to match between two people who aren't really the same person than missing a match between two people who are actually the same person
    - Minimize false positive

And Some Lessons to Learn

- Labeling data for training
  - "I actually labeled data myself. I extracted cases in which there is a match (e.g. Jennifer Williams / Jenny Williams; labeled as 1), cases that are a "close" match (e.g. Don Anderson / Daniel Anderson; labeled as 0), and added a large random sample from the data for labeling. The "close" matches allowed me to build a robust model that can differentiate very well between real matches and matches that are close but not actually matches. This wasn't quite a pleasure, but it made this project feasible :)"
- Human in the (painful) loop
  - "After running the optimized model for the first time I got a precision score of 0.85 on the test set"
  - "I will tell the model again and again that it was wrong. I took a vast amount of cases in which the age is similar, and also one of the names is the same"

What is Active Learning?