Philipps Universität Marburg

Lakehouse – Data Integration:
Data Profiling – Binary Datatype Analysis

Kevin Gieswein

Motivation

- **Finding meta-data** for individual data in big data sets

- **Data classification**

  – In data types

  – Binary data types separate data sets into two semantic classes.

- Using datatype to **unify data formats**

  – Unify float value format

  – Unify date format

  – In Binary data types difficult (Boolean VS. Binary)

    - {('True','False'), ('1','0'), ('Yes','No'),('t', 'f')} → (true, false)

    - {('Men', 'Women'), ('good', 'average'), ('free','paid'), ('Missed Shot', 'Made Shot')
          ('owned by a public authority', 'not owned by a public authority')} → ???

Related Work

- Synthesizing Type-Detection Logic for Rich Semantic Data Types using Open-source Code
  - Input: set of positive examples for a target data type and a search keyword (the target data type)
  - Output: synthesized type-detection function
    - 27.05.2018
    - https://dl.acm.org/doi/abs/10.1145/3183713.3196888
- DCoM: A Deep Column Mapper for Semantic Data Type Detection
  - Neural network fed with raw instance data
    - Release: 26.06.2021
    - Source: https://arxiv.org/pdf/2106.12871.pdf
- Sherlock: A Deep Learning Approach to Semantic Data Type Detection
  - Extracting features from raw data set to feed the neural network
    - Release: 25.07.2019
    - Source: https://dl.acm.org/doi/abs/10.1145/3292500.3330993

Definitions

- Binary data type:
  - A set of two values
  - Values are in a common top-level semantic data type which can be divided into two groups
    - Examples: gender (male, female),
      truth-values (true, false),
      ownerships (owned by X, not owned by X) ...
  - Difficulty:
    - Sometimes the top-level semantic data type can be divided into more than two groups but in special contexts some data set use only two which makes them sort of binary.
      - Examples: evaluations (good, bad)   {average, normal, OK, …},
        colours (black, white)   {green, yellow, red, …},
        categories (A, B) …   {C, D, E, X, …} .

Kevin Gieswein

Our Dataset

- Data:
  - Source: https://gittables.github.io/
  - Count: 1018140 CSV tables
  - Size: 19.8GB

- Making Dataset smaller by removing tables that are too empty
  - If more than 50% of the entries of a table are null values the table will be removed
    - 42.411 tables were deleted
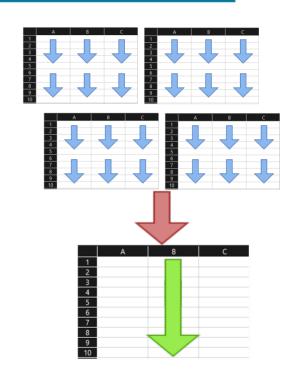   - Count after deleting: 975.729 CSV tables
    - 19.1GB

Our Method to counter the problem

1.) Identify binary value candidates.

2.) Define heuristics to rank the candidates.

3.) Compute a score which represents the rank of being a binary value

4.) Compute a rank which represents the probability of being a binary value

5.) Save our results in a CSV table

Identify binary value candidates

- To identify our candidates, we simply iterate through our data.

  - If an attribute converted to a set has only two values, we consider it as a candidate

  - Candidates will be saved as a tuple in a new CSV table

    - Example attribute in a table:
      - G = (male, male, female, male)
      - Set(G) = {male, female}
      - |Set(G)| = 2 → G is a candidate

Kevin Gieswein
Folie: 7

Heuristics to classify the candidates

1) The **Number of occurrences** of a candidate could increase the probability to be binary

- ('comment', 'story') : 316323
- ('1', '0') : 111918
- ('5.0', '2.0') : 53
- ('Y', 'N') : 8990
- ('Logged Out', 'Logged In') : 594
- ('neg', 'pos') : 53

2) If the two values of a candidate set have the **same data type** the probability could increase

- ('true', 'false') : (string, string)
- (1, -1) : (integer, integer)
- ('-', '+') : (string, string)
- ('-', 18.4) : (string, float)
- ('-', 1) : (string, integer)

Kevin Gieswein
Folie: 8

Heuristics to classify the candidates

3) If the two values of a candidate set have a good **jaccard similarity** the probability could increase

- sim('Female', 'Male') = 0.8

- sim('owned by a public authority', 'not owned by a public authority') = 0.9

- sim('Men', 'Women') = 0.6

- sim('1', '-1') = 0.5

- sim('Static', 'Dynamic') = 0.333

- sim('Yes','No') = sim('t','f') = sim('0','1') = 0

Kevin Gieswein
Folie: 9

Heuristics to classify the candidates

4) If one of the two values of a candidate set **occurs in another set** the probability could decrease

- ('imports', 'exports') : False
- ('Logged Out', 'Logged In') : False
- ('Missed Shot', 'Made Shot') : False
- ('www.google.com', 'http://www.blank.ne.jp') : False
- ('English', 'Japanese') : True
- ('True', 'False'); ('True','1'); ('', 'True') : True
- ('No', 'Yes'); ('Yes', 'NO'); ('Yes', 'Developable'), (' No', 'Yes') : True

Kevin Gieswein
Folie: 10

Candidate Scheme

- Considering all heuristics our generated candidates CSV table has the following scheme:

  - (binary_candidate, count, datatypes, jaccard_similarity, multiple_occurrence, final_score, rank)

    - Example Entries:
      - **('comment', 'story')**, 316323, ('string','string'), 0.222, True, 0.8222, 0.99
      - **('1', '0')**, 111918, ('integer', 'integer'), 0.0, True, 0.4769, 0.99

Score Calculation

```python
def calculate_score(attribute1: int, attribute2: bool, attribute3: float, attribute4: bool,
                    weighting: list,MAX_COUNT: int) -> float:
    SCORE_DIVISOR = 12
    scaled_attribute1 = attribute1 / MAX_COUNT * weighting[0]        # Bonus for Attribute 1
    scaled_attribute2 = 0.01 * weighting[1] if attribute2 else 0.0   # Bonus for Attribute 2
    scaled_attribute3 = attribute3 * weighting[2]                    # Bonus for Attribute 3
    scaled_attribute4 = -0.01 * weighting[3] if attribute4 else 0.0  # Penalty for Attribute 4
    total = scaled_attribute1 + scaled_attribute2 + scaled_attribute3 + scaled_attribute4
    score = total / SCORE_DIVISOR
    return score

# attribute1 : count

# attribute2 : equal data type

# attribute3 : jaccard similarity

# attribute4 : multiple occurrence

# weighting : [10, 1, 0.1, 0.1]
```

Kevin Gieswein
Folie: 12

Results

	A	B	C	D	E	F
1	binary_candidate	count	datatypes	jaccard_similarity	multiple_occurrence	final_score
2	('comment', 'story')	316323	('string', 'string')	0.2222222222222	True	0.835935185185185
3	('1', '0')	111918	('integer', 'integer')	0	True	0.295591032741849
4	('A', 'B')	29815	('string', 'string')	0	True	0.0792957691452513
5	('True', 'False')	24619	('string', 'string')	0.125	True	0.0666488959333761
6	('2', '1')	21882	('integer', 'integer')	0	True	0.0583967724446215
7	('0.0', '1.0')	9301	('float', 'float')	0.6666666666667	True	0.0308084634482264
8	('2', '0')	9435	('integer', 'integer')	0	True	0.0256059225854585
9	('imports', 'exports')	7423	('string', 'string')	0.5555555555556	True	0.0249350629472617
10	('Y', 'N')	8990	('string', 'string')	0	True	0.0244335976728428
11	('"\\""', \"DNP - Coach\\\'s Decision "\")	8558	('string', 'string')	0	True	0.0232955204543036
12	('1', '-1')	5016	('integer', 'integer')	0.5	True	0.0181310077041505
13	('0', '3')	5116	('integer', 'integer')	0	True	0.0142277848380716
14	('good', 'average')	4609	('string', 'string')	0.1428571428571	True	0.0140826009595677
15	('variety', 'species')	3766	('string', 'string')	0.2	True	0.0123379601968031
16	('3', '4')	4026	('integer', 'integer')	0	True	0.0113562474116647
17	('Men', 'Women')	2092	('string', 'string')	0.6	True	0.0112612443082967
18	('people', 'person')	2557	('string', 'string')	0.4285714285714	True	0.0110576865524585
19	('1.0', '-1.0')	1411	('float', 'float')	0.75	True	0.0107171920262938
20	('owned by a public authority', 'not owned by a public authority')	344	('string', 'string')	1	True	0.00998958000735535
21	('Yes', 'No')	3507	('string', 'string')	0	True	0.009988974086614
22	('101', '100')	302	('integer', 'integer')	1	True	0.00987893361110848
23	('model_purchase_requisition', 'model_purchase_requisition_line')	220	('string', 'string')	1	True	0.00966290969462648
24	('Static', 'Dynamic')	2328	('string', 'string')	0.3333333333333	True	0.00966074945546167
25	('0.0', '2.0')	1270	('float', 'float')	0.6666666666667	True	0.00965129182302056
26	('11.0', '10.0')	196	('float', 'float')	1	True	0.00959968318248541
27	('base.group_user', 'base.group_hr_user')	452	('string', 'string')	0.9166666666667	True	0.00957965486754573
28	('10', '100')	179	('integer', 'integer')	1	True	0.00955489773638549
29	('Das and Petrov (2011)', '(Das and Petrov 2011)')	166	('string', 'string')	1	True	0.00952065004230908
30	('1610612751', '1610612755')	148	('integer', 'integer')	1	True	0.00947323015820327
31	('1610612744', '1610612747')	146	('integer', 'integer')	1	True	0.00946796128219152
32	('1610612742', '1610612746')	143	('integer', 'integer')	1	True	0.00946005796817388
33	('1610612744', '1610612740')	138	('integer', 'integer')	1	True	0.00944688577814449
34	('1610612744', '1610612746')	137	('integer', 'integer')	1	True	0.00944425134013861
35	('1610612756', '1610612757')	137	('integer', 'integer')	1	True	0.00944425134013861
36	('1610612751', '1610612752')	136	('integer', 'integer')	1	True	0.00944161690213274
37	('1610612762', '1610612760')	136	('integer', 'integer')	1	True	0.00944161690213274
38	('1610612755', '1610612752')	133	('integer', 'integer')	1	True	0.0094337135881151
39	('1610612742', '1610612740')	133	('integer', 'integer')	1	True	0.0094337135881151
40	('1610612742', '1610612744')	133	('integer', 'integer')	1	True	0.0094337135881151

Philipps Universität Marburg

Kevin Gieswein
Folie: 13

Special Filter

```python
if useSpecialFilter:
    if "1610" in str(v[0][0]) or "1610" in str(v[0][1]):        # string often occurred
        probability = 0
    if "www." in str(v[0][0]) or "www." in str(v[0][1]):        # string is a link
        probability = 0
    if "http" in str(v[0][0]) or "http" in str(v[0][1]):        # string is a link
        probability = 0
    if len(str(v[0][0])) > 200 or len(str(v[0][1])) > 100:      # string is to long
        probability = 0
    if str(v[0][0]).lower() == str(v[0][1]).lower():            # equal string, example ('Victoria', 'VICTORIA')
        probability = 0

    for year in range(1800, 2023):                              # string contained a year (date data type)
        if str(year) in str(v[0][0]) or str(year) in str(v[0][1]):
            probability = 0

# v : current entry of the candidates CSV table
```
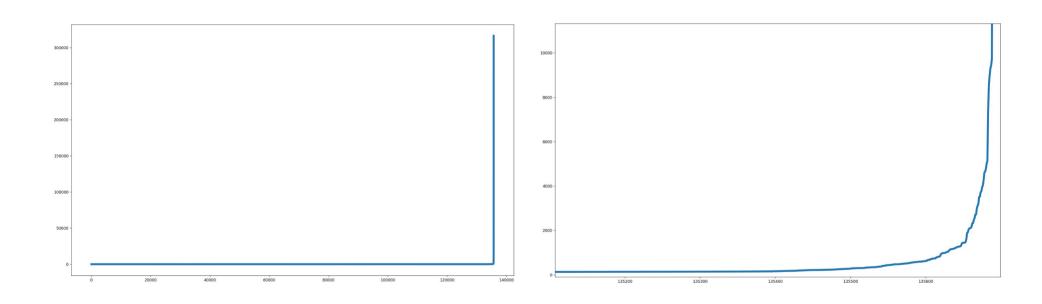
Kevin Gieswein
Folie: 14

Results - Score

	A	B	C	D	E	F
1	binary_candidate	count	datatypes	jaccard_similarity	multiple_occurrence	final_score
2	('comment', 'story')	316323	('string', 'string')	0.22222222222222	True	0.835935185185185
3	('1', '0')	111918	('integer', 'integer')	0	True	0.295591032741849
4	('A', 'B')	29815	('string', 'string')	0	True	0.0792957691452513
5	('True', 'False')	24619	('string', 'string')	0.125	True	0.0666488959333761
6	('2', '1')	21882	('integer', 'integer')	0	True	0.0583967724446215
7	('0.0', '1.0')	9301	('float', 'float')	0.66666666666667	True	0.0308084634482264
8	('2', '0')	9435	('integer', 'integer')	0	True	0.0256059225854585
9	('imports', 'exports')	7423	('string', 'string')	0.55555555555556	True	0.0249350629472617
10	('Y', 'N')	8990	('string', 'string')	0	True	0.0244335976728428
11	('"\""', '\"DNP - Coach\\\'s Decision "\")	8558	('string', 'string')	0	True	0.0232955204543036
12	('1', '-1')	5016	('integer', 'integer')	0.5	True	0.0181310077041505
13	('0', '3')	5116	('integer', 'integer')	0	True	0.0142277848380716
14	('good', 'average')	4609	('string', 'string')	0.14285714285714	True	0.0140826009595677
15	('variety', 'species')	3766	('string', 'string')	0.2	True	0.0123379601968031
16	('3', '4')	4026	('integer', 'integer')	0	True	0.0113562474116647
17	('Men', 'Women')	2092	('string', 'string')	0.6	True	0.0112612443082967
18	('people', 'person')	2557	('string', 'string')	0.42857142857143	True	0.0110576865524585
19	('1.0', '-1.0')	1411	('float', 'float')	0.75	True	0.0107171920262938
20	('owned by a public authority', 'not owned by a public authority')	344	('string', 'string')	1	True	0.00998958000735535
21	('Yes', 'No')	3507	('string', 'string')	0	True	0.009988974086614
22	('101', '100')	302	('integer', 'integer')	1	True	0.00987893361110848
23	('model_purchase_requisition', 'model_purchase_requisition_line')	220	('string', 'string')	1	True	0.00966290969462648
24	('Static', 'Dynamic')	2328	('string', 'string')	0.33333333333333	True	0.00966074945546167
25	('0.0', '2.0')	1270	('float', 'float')	0.66666666666667	True	0.00965129182302056
26	('11.0', '10.0')	196	('float', 'float')	1	True	0.00959968318248541
27	('base.group_user', 'base.group_hr_user')	452	('string', 'string')	0.91666666666667	True	0.00957965486754573
28	('10', '100')	179	('integer', 'integer')	1	True	0.00955489773638549
29	('base.group_sale_manager', 'base.group_sale_salesman')	116	('string', 'string')	1	True	0.00938892814201518
30	('access_sale_order_line_manufacturing_user', 'access_sale_order_manufacturing_user')	111	('string', 'string')	1	True	0.0093757559519859
31	('sale.order manufacturing.user', 'sale.order.line manufacturing.user')	111	('string', 'string')	1	True	0.0093757559519859
32	('app-supplychain-config-form', 'supply-chain-config-form')	34	('string', 'string')	1	True	0.00917290422553319
33	('Method_Citation,Hypothesis_Citation,Aim_Citation,Implication_Citation', 'Method_Citation,Hypothesis_Citation,Implication_Citation')	34	('string', 'string')	1	True	0.00917290422553319
34	('1', '11')	31	('integer', 'integer')	1	True	0.00916500091151555
35	('13.043478260869565', '8.695652173913043')	29	('float', 'float')	1	True	0.0091597320355038
36	('Not owned by a public authority', 'Owned by a public authority')	25	('string', 'string')	1	True	0.00914919428348028
37	('com.axelor.apps.purchase.db.PurchaseRequest', 'com.axelor.apps.purchase.db.PurchaseRequestCreator')	21	('string', 'string')	1	True	0.00913865653145677
38	('model_base_action_rule_lead_test', 'model_base_action_rule')	17	('string', 'string')	1	True	0.00912811877943326
39	('sales_team.group_sale_manager', 'sales_team.group_sale_salesman')	17	('string', 'string')	1	True	0.00912811877943326
40	('nz_21310', 'nz_21330')	16	('string', 'string')	1	True	0.00912548434142738
41	('link_tracker.model_link_tracker_code', 'link_tracker.model_link_tracker_click')	16	('string', 'string')	1	True	0.00912548434142738
42	('access_pos_multi_session_sync_pos', 'access_pos_multi_session_sync_multi_session')	16	('string', 'string')	1	True	0.00912548434142738
43	('perm.suppliermanagement.PurchaseOrderSupplierLine.rwc', 'perm.suppliermanagement.PurchaseOrderSupplierLine.rwcde')	15	('string', 'string')	1	True	0.0091228499034215
44	('2.5944828711436', '3.5944828711436')	14	('float', 'float')	1	True	0.00912021546541563
45	('perm.client.portal.OnlinePaymentMethod;rwcde', 'perm.client.portal.OnlinePaymentMethod;rwc')	14	('string', 'string')	1	True	0.00912021546541563
46	('1.0', '0.1')	12	('float', 'float')	1	True	0.00911494658940387
47	('asset.model_asset_asset', 'asset.model_asset_state')	12	('string', 'string')	1	True	0.00911494658940387
48	('-84.5351 29.0149 100000 2', '-84.5351 29.0149 100000 1')	12	('string', 'string')	1	True	0.00911494658940387
49	('Undergraduate Student', 'Graduate Student')	11	('string', 'string')	1	True	0.00911231215139799

Kevin Gieswein
Folie: 15

Count - Graph

Kevin Gieswein
Folie: 16

Results - Rank

binary_candidate	count	datatypes	jaccard_similarity	multiple_occurrence	final_score	rank
('1', '0')	111918	('integer', 'integer')	0	True	0.295591032741849	0.999985884478573
('A', 'B')	29815	('string', 'string')	0	True	0.0792957691452513	0.999971768957145
('True', 'False')	24619	('string', 'string')	0.125	True	0.0666488959333761	0.999957653435718
('2', '1')	21882	('integer', 'integer')	0	True	0.0583967724446215	0.999943537914291
('0.0', '1.0')	9301	('float', 'float')	0.66666666666667	True	0.0308084634482264	0.999929422392863
('2', '0')	9435	('integer', 'integer')	0	True	0.0256059225854585	0.999915306871436
('imports', 'exports')	7423	('string', 'string')	0.55555555555556	True	0.0249350629472617	0.999901191350008
('Y', 'N')	8990	('string', 'string')	0	True	0.0244335976728428	0.999887075828581
('"\\\\\\"', '\\\"DNP - Coach\\\\\\'s Decision "\\\")	8558	('string', 'string')	0	True	0.0232955204543036	0.999872960307154
('1', '-1')	5016	('integer', 'integer')	0.5	True	0.0181310077041505	0.999858844785726
('0', '3')	5116	('integer', 'integer')	0	True	0.0142277848380716	0.999844729264299
('good', 'average')	4609	('string', 'string')	0.14285714285714	True	0.0140826009595677	0.999830613742872
('variety', 'species')	3766	('string', 'string')	0.2	True	0.0123379601968031	0.999816498221444
('3', '4')	4026	('integer', 'integer')	0	True	0.0113562474116647	0.999802382700017
('Men', 'Women')	2092	('string', 'string')	0.6	True	0.0112612443082967	0.99978826717859
('people', 'person')	2557	('string', 'string')	0.42857142857143	True	0.0110576865524585	0.999774151657162
('1.0', '-1.0')	1411	('float', 'float')	0.75	True	0.0107171920262938	0.999760036135735
('owned by a public authority', 'not owned by a public authority')	344	('string', 'string')	1	True	0.0099898958000735535	0.999745920614307
('Yes', 'No')	3507	('string', 'string')	0	True	0.00998897408614	0.99973180509288
('101', '100')	302	('integer', 'integer')	1	True	0.00987893361110848	0.999717689571453
('model_purchase_requisition', 'model_purchase_requisition_line')	220	('string', 'string')	1	True	0.00966290969462648	0.999703574050025
('Static', 'Dynamic')	2328	('string', 'string')	0.33333333333333	True	0.00966074945546167	0.999689458528598
('0.0', '2.0')	1270	('float', 'float')	0.66666666666667	True	0.00965129182302056	0.999675343007171
('11.0', '10.0')	196	('float', 'float')	1	True	0.00959968318248541	0.999661227485743
('base.group_user', 'base.group_hr_user')	452	('string', 'string')	0.91666666666667	True	0.00957965486754573	0.999647111964316
('10', '100')	179	('integer', 'integer')	1	True	0.00955489773638549	0.999632996442889
('base.group_sale_manager', 'base.group_sale_salesman')	116	('string', 'string')	1	True	0.00938892814201518	0.999618880921461
('access_sale_order_line_manufacturing_user', 'access_sale_order_manufacturing_user')	111	('string', 'string')	1	True	0.00937575595198579	0.999604765400034
('sale.order manufacturing.user', 'sale.order.line manufacturing.user')	111	('string', 'string')	1	True	0.00937575595198579	0.999590649878607
('app-supplychain-config-form', 'supply-chain-config-form')	34	('string', 'string')	1	True	0.00917290422553319	0.999576534357179
('Method_Citation,Hypothesis_Citation,Aim_Citation,Implication_Citation', 'Method_Citation,Hypothesis_Citation,Implication_Citatio...	34	('string', 'string')	1	True	0.00917290422553319	0.999562418835752
('1', '11')	31	('integer', 'integer')	1	True	0.00916500091151555	0.999548303314324
('13.043478260869565', '8.695652173913043')	29	('float', 'float')	1	True	0.00915973203355038	0.999534187792897
('Not owned by a public authority', 'Owned by a public authority')	25	('string', 'string')	1	True	0.00914919428348028	0.99952007227147
('com.axelor.apps.purchase.db.PurchaseRequest', 'com.axelor.apps.purchase.db.PurchaseRequestCreator')	21	('string', 'string')	1	True	0.00913865653145677	0.999505956750042
('model_base_action_rule_lead_test', 'model_base_action_rule')	17	('string', 'string')	1	True	0.00912811877943326	0.999491841228615
('sales_team.group_sale_manager', 'sales_team.group_sale_salesman')	17	('string', 'string')	1	True	0.00912811877943326	0.999477725707188
('nz_21310', 'nz_21330')	16	('string', 'string')	1	True	0.00912548434142738	0.99946361018576
('link_tracker.model_link_tracker_code', 'link_tracker.model_link_tracker_click')	16	('string', 'string')	1	True	0.00912548434142738	0.999449494664333
('access_pos_multi_session_sync_pos', 'access_pos_multi_session_sync_multi_session')	16	('string', 'string')	1	True	0.00912548434142738	0.999435379142906
('perm.suppliermanagement.PurchaseOrderSupplierLine.rwc', 'perm.suppliermanagement.PurchaseOrderSupplierLine.rwcde')	15	('string', 'string')	1	True	0.0091228499034215	0.999421263621478
('2.5944828711436', '3.5944828711436')	14	('float', 'float')	1	True	0.00912021546541563	0.999407148100051
('perm.client.portal.OnlinePaymentMethod;rwcde', 'perm.client.portal.OnlinePaymentMethod;rwc')	14	('string', 'string')	1	True	0.00912021546541563	0.999393032578623
('1.0', '0.1')	12	('float', 'float')	1	True	0.00911494658940387	0.999378917057196
('asset.model_asset_asset', 'asset.model_asset_state')	12	('string', 'string')	1	True	0.00911494658940387	0.999364801535769
('-84.5351 29.0149 100000 2', '-84.5351 29.0149 100000 1')	12	('string', 'string')	1	True	0.00911494658940387	0.999350686014341
('Undergraduate Student', 'Graduate Student')	11	('string', 'string')	1	True	0.00911231215139799	0.999336570492914

Kevin Gieswein
Folie: 17

Results

- Found **135696 possible candidates** for binary attributes

- **64851** candidates got a **score of 0**

- We **lost 47,7%** of our candidates and sorted them by score

- Not perfect:
  - rank('neg','pos') = 0.055 = 5.5%
  - rank('y','n') = 0.054 = 5.4%
  - rank('18+','Juniors') = 0.01 = 1%
  - rank('imports','exports') = 0.99 = 99%
  - rank('Static', 'Dynamic') = 0.99 = 99%

Future Works

- Look for semantic data types

  - Answers : {"yes", "no"}

  - Evaluations : {"good", "bad"}

  - Negations : {"X", "not X"}, {"positive", "negative"}

- Analyse semantic data types instead of just integer, float and strings like in heuristic_2

Summary

`git clone https://github.com/JoseroKies/Data_Profiling_Binary_Datatype_Analysis.git`

binary_candidate	count	datatypes	jaccard_similarity	multiple_occurrence	final_score	rank
('1', '0')	111918	('integer', 'integer')	0	True	0.295591032741849	0.999985884478573
('A', 'B')	29815	('string', 'string')	0	True	0.0792957691452513	0.999971768957145
('True', 'False')	24619	('string', 'string')	0.125	True	0.0666488959333761	0.999957653435718
('2', '1')	21882	('integer', 'integer')	0	True	0.0583967724446215	0.999943537914291
('0.0', '1.0')	9301	('float', 'float')	0.66666666666667	True	0.0308084634482264	0.999929422392863
('2', '0')	9435	('string', 'string')	0	True	0.0256059225854585	0.999915306871436
('imports', 'exports')	7423	('string', 'string')	0.55555555555556	True	0.0249350629472617	0.9999011913350008
('Y', 'N')	8990	('string', 'string')	0	True	0.0244335976728428	0.999887075828581
('"\\\\\\"', '\\\\"DNP - Coach\\\\\\\\'s Decision "\\\\')	8558	('string', 'string')	0	True	0.0232955204543036	0.999872960307154
('1', '-1')	5016	('string', 'string')	0.5	True	0.0181310077041505	0.999858844785726
('0', '3')	5116	('integer', 'integer')	0	True	0.0142277848380716	0.999844729264299
('good', 'average')	4609	('string', 'string')	0.14285714285714	True	0.0140826009595677	0.999830613742872
('variety', 'species')	3766	('string', 'string')	0.2	True	0.0123379601968031	0.999816498221444
('3', '4')	4026	('integer', 'integer')	0	True	0.0113562474116647	0.999802382700017
('Men', 'Women')	2092	('string', 'string')	0.6	True	0.0112612443082967	0.99978826717859
('people', 'person')	2557	('string', 'string')	0.42857142857143	True	0.0110576865524585	0.999774151657162
('1.0', '-1.0')	1411	('float', 'float')	0.75	True	0.0107171920262938	0.999760036135735
('owned by a public authority', 'not owned by a public authority')	344	('string', 'string')	1	True	0.00998958000735535	0.999745920614307
('Yes', 'No')	3507	('string', 'string')	0	True	0.00998894074086614	0.99973180509288
('101', '100')	302	('integer', 'integer')	1	True	0.00987893361110848	0.999717689571453
('model_purchase_requisition', 'model_purchase_requisition_line')	220	('string', 'string')	1	True	0.00966290969462648	0.9997035740500025
('Static', 'Dynamic')	2328	('string', 'string')	0.33333333333333	True	0.00966074945546167	0.999689458528598
('0.0', '2.0')	1270	('float', 'float')	0.66666666666667	True	0.00965129182302056	0.999675343007171
('11.0', '10.0')	196	('float', 'float')	1	True	0.00959968318248541	0.999661227485743
('base.group_user', 'base.group_hr_user')	452	('string', 'string')	0.91666666666667	True	0.00957965486754573	0.999647111964316
('10', '100')	179	('integer', 'integer')	1	True	0.00955489773638549	0.999632996442889
('base.group_sale_manager', 'base.group_sale_salesman')	116	('string', 'string')	1	True	0.00938892814201518	0.999618880921440
('access_sale_order_line_manufacturing_user', 'access_sale_order_manufacturing_user')	111	('string', 'string')	1	True	0.00937575595198579	0.9996047654000034
('sale.order manufacturing.user', 'sale.order.line manufacturing.user')	111	('string', 'string')	1	True	0.00937575595198579	0.999590649878607
('app-supplychain-config-form', 'supply-chain-config-form')	34	('string', 'string')	1	True	0.00917290422553319	0.999576534357179
('Method_Citation,Hypothesis_Citation,Aim_Citation,Implication_Citation', 'Method_Citation,Hypothesis_Citation,Implication_Citation')	34	('string', 'string')	1	True	0.00917290422553319	0.999562418835752
('1', '11')	31	('integer', 'integer')	1	True	0.00916500091151555	0.999548303314324
('13.043478260869565', '8.695652173913043')	29	('float', 'float')	1	True	0.00915973203550338	0.999534187792897
('Not owned by a public authority', 'Owned by a public authority')	25	('string', 'string')	1	True	0.00914919428348028	0.99952007227147
('com.axelor.apps.purchase.db.PurchaseRequest', 'com.axelor.apps.purchase.db.PurchaseRequestCreator')	21	('string', 'string')	1	True	0.00913865653145677	0.999505956750042
('model_base_action_rule_lead_test', 'model_base_action_rule')	17	('string', 'string')	1	True	0.00912811877943326	0.999491841228615
('sales_team.group_sale_manager', 'sales_team.group_sale_salesman')	17	('string', 'string')	1	True	0.00912811877943326	0.999477725707188
('nz_21310', 'nz_21330')	16	('string', 'string')	1	True	0.00912548434142738	0.99946361018576
('link_tracker.model_link_tracker_code', 'link_tracker.model_link_tracker_click')	16	('string', 'string')	1	True	0.00912548434142738	0.999449494664333
('access_pos_multi_session_sync_pos', 'access_pos_multi_session_sync_multi_session')	16	('string', 'string')	1	True	0.00912548434142738	0.999435379142906
('perm.suppliermanagement.PurchaseOrderSupplierLine.rwc', 'perm.suppliermanagement.PurchaseOrderSupplierLine.rwcde')	15	('string', 'string')	1	True	0.00912284990034215	0.999421263621478
('2.5944828711436', '3.5944828711436')	14	('float', 'float')	1	True	0.00912021546541563	0.999407148100051
('perm.client.portal.OnlinePaymentMethod;rwcde', 'perm.client.portal.OnlinePaymentMethod;rwc')	14	('string', 'string')	1	True	0.00912021546541563	0.999393032578623
('1.0', '0.1')	12	('float', 'float')	1	True	0.00911494658940387	0.999378917057196
('asset.model_asset_asset', 'asset.model_asset_state')	12	('string', 'string')	1	True	0.00911494658940387	0.999364801535769
('-84.5351 29.0149 100000 2', '-84.5351 29.0149 100000 1')	12	('string', 'string')	1	True	0.00911494658940387	0.999350686014341
('Undergraduate Student', 'Graduate Student')	11	('string', 'string')	1	True	0.00911231215139799	0.999336570492914
('F', 'fF')	11	('string', 'string')	1	True	0.00911231215139799	0.999322454971487

Kevin Gieswein
Folie: 20