Agricultural Bank of China, New York Branch Project (ABCNY)

Machine Learning Algorithm to Detect String Matches and Automated Test Case Generator

Student Team

Bohao He (bohaohe16@gwu.edu)

Shuning Ma (shuningma@gwu.edu)

Wenyuan Zou (iriszou@gwu.edu)

Xuan Zhao (xuanzhao@gwu.edu)

Yingying Liu (liuyy0201@gwu.edu)

Yuqi Wu (wyuqi30@gwu.edu)

Clients

Andrew McAdams (mcadams.andrew@outlook.com)

Matthew Finan (matthewfinan@abchinausa.com)

Neelansh Prasad (neelanshprasad@abchinausa.com)

Business Analytics Practicum (DNSC 6317)

Professor: Brian Murrow (brianmurrow@gwu.edu)

Due Date: December 09, 2022

Table of Contents

Executive Summary

Agricultural Bank of China, New York Branch (“ABCNY”) and a student team built a machine learning algorithm that classifies two strings as matches or not matches. The classification algorithm takes two names, an input name and a reference name, and classifies the two strings as match or no match.
To build the algorithm, the student team first created an automated test case name generator to automatically generate hundreds of name variations based on a Test Case Type Tracker provided by the client.
The student team then developed a machine learning algorithm that classified the input name and reference name as matches or not matches. The client indicated the algorithm would be used to weed out the obvious false positives generated by the ABCNY’s name matching system.

Problem Understanding

Business Objectives

Generate test case types data by Python to provide it to clients. Then confirm bad actors matching, as well as the confusion matrix based on the machine learning model. Increase the degree of accuracy for matching correctly along with the OFAC sanctions list.

Assess Overall Situation

1. Project Requirement

1) Technical Demand: Generate an automated test case generator in Python that creates name variations of actual names for our testing, we need to acquire name scores for ABCNY’s name matching system.
2) Business Demand: Eventually figure out the bank's fuzzy string-matching system to increase the degree of accuracy when employees want to check a specific name in financial background information.

2. Assess Risks

Both of the following risk types will reduce the accuracy of the financial system, which could cause a certain probability to affect the trust of the bank's financial system.

1) False Negative: The risk will lead to the entities such as sanctioned individuals or companies to escape from the financial system and lead to potential financial crime risks.

2) False Positive: The risk will cause an unnecessary financial cost for those individuals, companies, and vessels entities who are not on the sanctions list. That's why we want to weed out false positives.

3. Contingencies

1) Subjective: Hacker intrusion
2) Objective: Catastrophic events

4. Cost-Benefit Analysis

Manpower maintenance like model revising and version upgrading when the project done. But a mature fuzzy string matching system can dramatically enhance a bank's work efficiency. At the same time, it can also reduce the probability of society's financial crimes such as tax evasion, embezzlement of company funds, etc.

○ Determine Data Processing Goals

In order to achieve the business objectives, to determine data processing goals is one of the critical processes. At the very beginning, each of us will spend time generating an automated test case generator. After we accomplished fuzzy string-matching tasks, we wrote out machine learning models so as to train test case prediction along with the sanctions list to weed out false positives.

● Methodology

1. As part of this project, the team generated an automated test case generator in Python that created name variations of actual names for our testing.
2. The student team then test name variations and created an algorithm using Machine Learning, which classifies strings as matches or not matches.

○ Data Analyzed

Data Source

1. OFAC list: U.S. Department of The Treasury website

	0	1	2	3	4	5	6	7	8	9	10	11
0	36	AEROCARIBBEAN AIRLINES	-0-	CUBA	-0-	-0-	-0-	-0-	-0-	-0-	-0-	-0-
1	173	ANGLO-CARIBBEAN CO., LTD.	-0-	CUBA	-0-	-0-	-0-	-0-	-0-	-0-	-0-	-0-
2	306	BANCO NACIONAL DE CUBA	-0-	CUBA	-0-	-0-	-0-	-0-	-0-	-0-	-0-	a.k.a. 'BNC'.
3	424	BOUTIQUE LA MAISON	-0-	CUBA	-0-	-0-	-0-	-0-	-0-	-0-	-0-	-0-
4	475	CASA DE CUBA	-0-	CUBA	-0-	-0-	-0-	-0-	-0-	-0-	-0-	-0-
...	...	...	...	...	...	...	...	...	...	...	...	...
10563	39257	ABNOUSH, Salar	individual	IRAN-HR	-0-	-0-	-0-	-0-	-0-	-0-	-0-	DOB 02 May 1962; POB Hamedan, Iran; nationalit...
10564	39258	IRAN'S MORALITY POLICE	-0-	IRAN-HR	-0-	-0-	-0-	-0-	-0-	-0-	-0-	Additional Sanctions Information - Subject to ...
10565	39259	MIRZAEI, Haj Ahmad	individual	IRAN-HR	Colonel	-0-	-0-	-0-	-0-	-0-	-0-	DOB 09 Feb 1957; nationality Iran; Additional ...
10566	39260	ROSTAMI CHESHMEH GACHI, Mohammad	individual	IRAN-HR	General	-0-	-0-	-0-	-0-	-0-	-0-	DOB 1976 to 1977; nationality Iran; Additional...
10567	□	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

10568 rows × 12 columns

2. Automated Test Case Generator: 515 types of name variations generated from student team members

	UID	Theme	Category	Sub-category	Entity-Type	Test Case ID	OFAC List UID	Original Name	Test Case Name
0	UID-212	Names where name parts are Modified	Character replaced by Number and Special Chara...	1 Letter replaced by number and 1 letter repla...	Individual	UID-212 - 9105	34658	FRADKOV, Petr Mikhailovich	F1ADKOV, Petr Mikhailovi()h
1	UID-212	Names where name parts are Modified	Character replaced by Number and Special Chara...	1 Letter replaced by number and 1 letter repla...	Individual	UID-212 - 7000	26683	ALVARES, Carlos	ALVARE6, C#rlos
2	UID-212	Names where name parts are Modified	Character replaced by Number and Special Chara...	1 Letter replaced by number and 1 letter repla...	Individual	UID-212 - 600	7828	AMDOUNI, Mehrez	AM(1OUNI, Mehrez
3	UID-212	Names where name parts are Modified	Character replaced by Number and Special Chara...	1 Letter replaced by number and 1 letter repla...	Individual	UID-212 - 6690	26078	TUBAIGY, Salah Muhammed A.	1UBAIGY, Salah Mu)ammed A.
4	UID-212	Names where name parts are Modified	Character replaced by Number and Special Chara...	1 Letter replaced by number and 1 letter repla...	Individual	UID-212 - 1100	9390	ALTAMIRANO LOPEZ, Hector	ALTAMIRANO 0OPEZ, Hec^or
5	UID-212	Names where name parts are Modified	Character replaced by Number and Special Chara...	1 Letter replaced by number and 1 letter repla...	Individual	UID-212 - 3678	16993	SCHIAVONE, Francesco	S(HIAVONE, Frances7o
6	UID-212	Names where name parts are Modified	Character replaced by Number and Special Chara...	1 Letter replaced by number and 1 letter repla...	Individual	UID-212 - 9508	35491	TSED, Nikolay Grigorevich	TSED, #ikolay Grigorevic1
7	UID-212	Names where name parts are Modified	Character replaced by Number and Special Chara...	1 Letter replaced by number and 1 letter repla...	Individual	UID-212 - 2197	12088	PELAEZ LOPEZ, John Jairo	PELAEZ LOPEZ, John Ja&7o
8	UID-212	Names where name parts are Modified	Character replaced by Number and Special Chara...	1 Letter replaced by number and 1 letter repla...	Individual	UID-212 - 6002	24519	AL-SAYYID, Ibrahim Amin	AL-SAYYID, Ibrah3m Am#n
9	UID-212	Names where name parts are Modified	Character replaced by Number and Special Chara...	1 Letter replaced by number and 1 letter repla...	Individual	UID-212 - 5363	22863	AL-MANSUR, Salim Mustafa Muhammad	AL-MANSUR, Salim Mustafa @uhamma8

3. Bridger Score: derived from ABCNY side's specific software

index	UID	Theme	Category	Sub-category	Entity-Type	Test Case ID	OFAC List UID	Original Name	Test Case Name	BRIDGER SCORE
18	UID-4	Positive Control	Exact Match	100% true match	Individual	UID-4 - 9547	35523	SIMIGIN, Pavel Vladimirovich	SIMIGIN, Pavel Vladimirovich	100
19	UID-4	Positive Control	Exact Match	100% true match	Individual	UID-4 - 2726	13480	GONZALEZ PARADA, Juvencio Ignacio	GONZALEZ PARADA, Juvencio Ignacio	100
20	UID-5	Name Additions	Articles	> 2 Articles added	Entity	UID-5 - 3564	16736	CHERNOMORNEFTEGAZ	AN CHERNOMORNEFTEGAZ AN A	100
21	UID-5	Name Additions	Articles	> 2 Articles added	Entity	UID-5 - 7762	29058	CASTLE HOLDING GMBH	OF CASTLE HOLDING GMBH A A	79
22	UID-5	Name Additions	Articles	> 2 Articles added	Entity	UID-5 - 9153	34753	MOBILNYE PLATEZHI LIMITED LIABILITY COMPANY	OR MOBILNYE PLATEZHI LIMITED LIABILITY COMPANY OF A	96
23	UID-5	Name Additions	Articles	> 2 Articles added	Entity	UID-5 - 352	7295	VISCAYA LTDA.	OF VISCAYA LTDA. AN OR	76
24	UID-5	Name Additions	Articles	> 2 Articles added	Entity	UID-5 - 1877	11477	CONSULTORIA EN CAMBIOS FALCON S.A. DE C.V.	A CONSULTORIA EN CAMBIOS FALCON S.A. DE C.V. OF OF	87
25	UID-5	Name Additions	Articles	> 2 Articles added	Entity	UID-5 - 6678	25978	GRANATURA, S. DE P.R. DE R.L. DE C.V.	A GRANATURA, S. DE P.R. DE R.L. DE C.V. OR A	72
26	UID-5	Name Additions	Articles	> 2 Articles added	Entity	UID-5 - 6136	25017	BONYAD TAAVON BASIJ	AN BONYAD TAAVON BASIJ OF OR	97
27	UID-5	Name Additions	Articles	> 2 Articles added	Entity	UID-5 - 5273	22488	TSMRBANK, OOO	OR TSMRBANK, OOO OF OF	80
28	UID-5	Name Additions	Articles	> 2 Articles added	Entity	UID-5 - 9102	34644	PUBLIC JOINT STOCK COMPANY ALROSA	A PUBLIC JOINT STOCK COMPANY ALROSA OF OR	80
29	UID-5	Name Additions	Articles	> 2 Articles added	Entity	UID-5 - 8326	30874	IRAN MOBIN ELECTRONIC DEVELOPMENT COMPAN	A IRAN MOBIN ELECTRONIC DEVELOPMENT COMPANY AN A	97
30	UID-7	Name Additions	Articles	1 Articles added	Entity	UID-7 - 8127	30372	RAH NEGAR PARS MIDDLE EAST COMPANY	RAH NEGAR PARS MIDDLE EAST COMPANY A	100
31	UID-7	Name Additions	Articles	1 Articles added	Entity	UID-7 - 1199	9571	AGROPECUARIA PALMA DEL RIO S.A.	AGROPECUARIA PALMA DEL RIO S.A. AN	96
32	UID-7	Name Additions	Articles	1 Articles added	Entity	UID-7 - 10874	39188	INTERNATIONAL CENTER FOR QUANTUM OPTICS AI	INTERNATIONAL CENTER FOR QUANTUM OPTICS AND QUAN	100
33	UID-7	Name Additions	Articles	1 Articles added	Entity	UID-7 - 8751	32895	BRAVERY MARITIME CORPORATION	BRAVERY MARITIME CORPORATION AN	96
34	UID-7	Name Additions	Articles	1 Articles added	Entity	UID-7 - 9833	35896	AKTSIONERNOE OBSHCHESTVO VERKHNEUFALEISKI	AKTSIONERNOE OBSHCHESTVO VERKHNEUFALEISKII ZAVOD	100
35	UID-7	Name Additions	Articles	1 Articles added	Entity	UID-7 - 306	7219	ASKATASUNA	ASKATASUNA AN	100
36	UID-7	Name Additions	Articles	1 Articles added	Entity	UID-7 - 7751	29042	DAMASCUS CHAM FOR MANAGEMENT LLC	DAMASCUS CHAM FOR MANAGEMENT LLC OF	100
37	UID-7	Name Additions	Articles	1 Articles added	Entity	UID-7 - 7092	26897	ORIENT CLUB	ORIENT CLUB OF	97
38	UID-7	Name Additions	Articles	1 Articles added	Entity	UID-7 - 1535	10575	BELNEFTEKHIM USA, INC.	BELNEFTEKHIM USA, INC. AN	96

Data Proportion

We set 50% matches and 50% no-matches to form a balanced dataset, therefore build a proper machine learning model.

1. 25% match (fuzzy match): 4627 rows from automated test case generator
2. 25% match (exact match): 4627 rows with totally same original name and test case name randomly selected from OFAC list
3. 50% no-match: 9254 rows with totally different original name and test case name randomly selected from OFAC list

- ### Analytics Techniques Used

The project was divided in three parts:

1. Pandas to create an automated test case generator to build and test a classification algorithm.
2. Create an algorithm using machine learning that determines if two strings are matches
3. Generated automated reports and visualizations in Python to support building the machine learning model

- ## Results, Conclusions, and Recommendations

- ### Machine Learning Algorithm Result

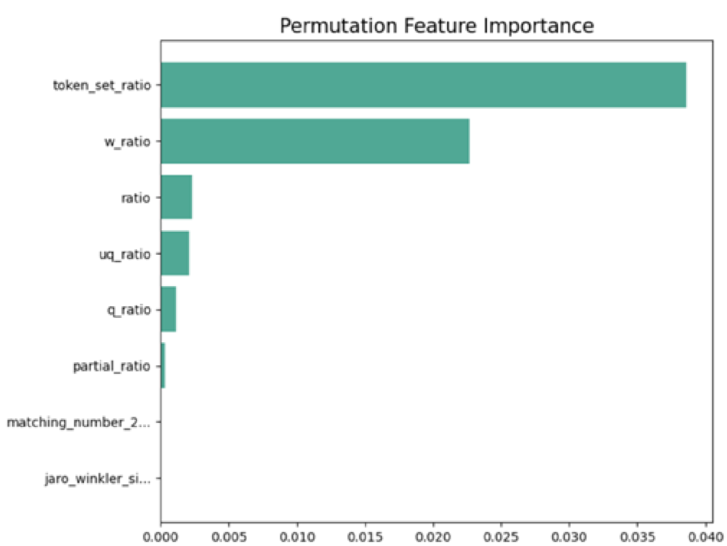
Feature Engineering

The 17 X features are generated using similarity functions with original names and test case names as inputs.

X Features	Description
levenshtein_distance	Levenshtein distance represents the number of insertions, deletions, and substitutions required to change one word to another.
damerau_levenshtein_distance	A modification of Levenshtein distance, Damerau-Levenshtein distance counts transpositions (such as ifsh for fish) as a single edit.
hamming_distance	Hamming distance is the measure of the number of characters that differ between two strings.
jaro_similarity	Jaro distance is a string-edit distance that gives a floating point response in [0,1] where 0 represents two completely dissimilar strings and 1 represents identical strings.
jaro_winkler_similarity	Jaro-Winkler is a modification/improvement to Jaro distance, like Jaro it gives a floating point response in [0,1] where 0 represents two completely dissimilar strings and 1 represents identical strings.
match_rating_comparison	The Match rating approach algorithm is an algorithm for determining whether or not two names are pronounced similarly.
ratio	Ratio function computes the standard Levenshtein distance similarity ratio between two sequences.
partial_ratio	Partial Ratio matches based on best substrings
token_sort_ratio	Token Sort Ratio tokenizes the strings and sorts them alphabetically before matching
token_set_ratio	Token Set Ratio tokenizes the strings and compared the intersection and remainder
w_ratio	W ratio handles lower and upper cases and some other parameters too
uq_ratio	UQ ratio is a unicode version of QRatio.
q_ratio	Q ratio method performs a quick ratio comparison between two strings. Runs full_process from utils on both strings. Short circuits if either of the strings is empty after processing.
matching_numbers	Matching numbers extracts numeric data from the names, i.e. 64, and looks at their level of similarity between the two names.
matching_numbers_log	Log version of matching_numbers
log_fuzz_score	Log version of (ratio + partial_ratio + token_sort_ratio + token_set_ratio)
log_fuzz_score_numbers	Log version of (fuzz_score + matching_numbers)

Feature Importance

The feature importance plot shown below was from ReLU-DNN model, our machine learning model with the best result. It was generated using a built-in visualization function of PiML package.



Validation Data Result

Charts below indicate that all 13 machine learning models performed well and all had high accuracy result for validation dataset.

	model	accuracy	mae	precision	recall	f1	roc	run_time	tp	fp	tn	fn
0	DummyClassifier_stratified	0.503671	0.496329	0.512557	0.522162	0.517315	0.503318	0.0	1449	1378	1295	1326
1	KNeighborsClassifier	0.994310	0.005690	0.997823	0.990991	0.994395	0.994373	0.01	2750	6	2667	25
2	XGBClassifier	0.993759	0.006241	0.994587	0.993153	0.993869	0.993771	0.06	2756	15	2658	19
3	DecisionTreeClassifier	0.989537	0.010463	0.986399	0.993153	0.989765	0.989468	0.0	2756	38	2635	19
4	RandomForestClassifier	0.994677	0.005323	0.996744	0.992793	0.994764	0.994713	0.02	2755	9	2664	20
5	AdaBoostClassifier	0.993025	0.006975	0.994937	0.991351	0.993141	0.993057	0.02	2751	14	2659	24
6	GradientBoostingClassifier	0.994310	0.005690	0.996023	0.992793	0.994405	0.994339	0.03	2755	11	2662	20
7	Perceptron	0.984031	0.015969	0.995941	0.972613	0.984139	0.984249	0.0	2699	11	2662	76
8	MLP	0.993759	0.006241	0.997098	0.990631	0.993854	0.993819	0.04	2749	8	2665	26
9	XGBClassifer tuned	0.993209	0.006791	0.992092	0.994595	0.993342	0.993182	0.01	2760	22	2651	15

Register ReLU-DNN Done

	ACC	AUC	Recall	Precision	F1
Train	0.9893	0.9974	0.9863	0.9923	0.9893
Test	0.9933	0.9982	0.9920	0.9945	0.9933
Gap	0.0041	0.0007	0.0057	0.0023	0.0040

Register GAMI-Net Done

	ACC	AUC	Recall	Precision	F1
Train	0.9920	0.9987	0.9882	0.9960	0.9921
Test	0.9951	0.9995	0.9938	0.9963	0.9951
Gap	0.0031	0.0008	0.0056	0.0004	0.0030

Register EBM Done

	ACC	AUC	Recall	Precision	F1
Train	0.9945	0.9995	0.9920	0.9971	0.9945
Test	0.9950	0.9993	0.9934	0.9963	0.9949
Gap	0.0004	-0.0002	0.0014	-0.0007	0.0004

Test Data Result and Model selection

ReLU-DNN had significantly larger number of true negatives than all the others, and only several more false negatives as exchange. Since we viewed the number of true negatives as the most important evaluation criteria of our classification models, we chose ReLU-DNN as our best machine learning model.

Test data	tp	fp
73860	3047	70813

Model	tp	fn	fp	tn
XGBClassifier tuned	2961	86	51366	19447
ReLU-DNN	2959	88	49767	21046
GAMI-Net	2964	83	52561	18252
EBM	2964	83	51030	19783

Loss	Improvement
2.82%	27.46%
2.89%	29.72%
2.72%	25.77%
2.72%	27.94%

- ## Overall Result and Conclusion

After creating our Machine Learning algorithm and training it on the test cases we generated with our automated test case generator, we used our machine learning name matching classification algorithm to increase the efficiency of the Branch's production system output by weeding out obvious false positives.

The Bank's production system Bridger created 73,860 possible matches during a review period. Our machine learning algorithm ingested those matches and accurately designated 21,046 of them as "Not Matches", while incorrectly designated 88 as "Not Matches" when they were actually "Matches".
As such, we were able to reduce the population of matches by 30% by classifying them accurately as false positives while creating false negatives for only 2.9% of the Matches population, which the Bank found to be a successful tradeoff.

● Potential Next Steps

○ Limitations and Improvements

1. **Limitation 1:** Limitation on language translation. Our data set contains lists of foreign names that might be very different in their original language but seem to be similar in English.

   Improvements:
   1) Translate letters in other languages into English.
   2) Recruit consultants who are native speakers of corresponding foreign languages.

2. **Limitation 2:** Data examples are too idealistic and do not accurately capture real world situations (e.g. Entity, Individual, Vessel's matching).

   Improvements:
   1) Reconstruct current data composition with building more realistic no-match name variations.
   2) Build more test case names with different name variations.

3. **Limitation 3:** Limitation on similarity metrics. As our similarity metrics are calculated from similarity distance, and the mathematical theory behind some of them would be similar, these

similarity metrics, as our X-features, probably are not comprehensive enough to do the classification.

Improvement:
Look for more similarity metrics which evaluate 2 strings with different principles.

- ### Monitoring and Maintaining

  Establish a weekly or monthly check system based on the log, which describes the problems during the running time.

Appendices Including Code Developed for The Project

Coding Documents Link: ABCNY Project Coding Documents

1. Automated Test Case Generator Coding
2. Machine Learning Algorithms Coding