Machine Learning For Detecting Nuclear-related Strategic Trade: An Introduction To Supervised Classification And NLP-assisted Matching For Transaction Identification

Christopher Nelson
Strategic Trade Control Research Group LLC

ABSTRACT

Detecting the transfer of nuclear-related strategic goods within the international trading system is a long-standing challenge. The small volume, difficulty correlating Harmonized System codes strategic trade control lists, and the dual-use nature of many of the commodities combine to make this a tremendous challenge. Advances in trade data collection and computing capabilities now provide the opportunity to apply machine learning to this issue. This paper provides an introduction to two techniques to improve the identification of nuclear-related strategic trade transactions: supervised classification and natural language processing (NLP) with fuzzy matching. The supervised classification approach uses data resampling and classification algorithms to model common patterns and characteristics that separate transactions involving nuclear-related strategic goods from broader international trade flows. This approach creates models using historical transaction-level and export licensing data in order to predict whether new transactions are likely to contain a nuclear-related strategic good. The second technique uses NLP to operationalize strategic trade control lists and other relevant descriptions of nuclear-related commodities. After pre-processing the text, this approach uses fuzzy matching to identify nuclear-related trade based on descriptions provided in shipping documentation. Both of these approaches can be used by state authorities to improve strategic trade control enforcement and outreach for nuclear-related commodities.

INTRODUCTION

Nuclear-related strategic goods are the materials, technology, and equipment that can be used in the development or delivery of nuclear weapons. These commodities are referred to strategic goods because they are subject to strategic trade controls (STCs), a term that encompasses not only licensing and regulation of these goods upon export, but also during import, re-export, transit, and transshipment. In general, nuclear-related strategic goods refers to those covered under the Nuclear Suppliers Group Trigger and Dual-use lists.[1] Most of these commodities are dual-use, i.e. they have both nuclear and non-nuclear applications. High strength steel, machine tools, heat exchangers, and mass spectrometers, to name a few, are all more commonly used outside of nuclear end-uses than for them.

States face several challenges in identifying international trade in these commodities. First, trade in nuclear-related strategic goods is a small portion of global trade, which makes detecting export control violations and illicit trade very difficult. STCs are just one of many missions of state authorities with regard to international trade. They have limited resources to evaluate a massive volume of transactions, not only for illicit trade in strategic goods, but for tariff evasion, border security, and other smuggling efforts. In addition, the nomenclatures to track trade generally and strategic goods specifically were designed with different fundamental objectives and therefore do not correlate well. Attempting to link export control classification numbers (ECCNs) for strategic goods with their Harmonized System (HS) codes is a challenge. The connections linking HS codes and ECCNs are rarely one-to-one. A strategic good could be shipped under a host of HS codes and a HS code could correlate to multiple ECCNs. There are also almost never technical specifications in HS code descriptions, which are the key delineating factor on strategic control lists. Finally, bad actors take advantage of the weaknesses in the international trading system to obfuscate their activities. They intentionally misclassify products, exploit transshipment to obscure the chain of control, and seek out trading partners that are uninformed about their export control obligations, among many other tactics.

The combination of these factors makes it a challenge for state authorities to detect potential illicit transfers of nuclear-related strategic goods. The primary advantage authorities have to combat these issues is the fact that international trade is a data-rich environment. The large amount of data collected in the course of normal trade and export licensing activities can be leveraged to develop advanced and adaptable detection models. From medicine to banking to internet security, many fields have embraced machine learning techniques to solve similar problems of identifying a small, but critical number of problem cases in large datasets. This paper proposes doing the same for detecting nuclear-related strategic goods in international trade transactions.

Machine learning uses computer calculated algorithms to identify patterns in large amounts of data. Machine learning allows the algorithms to make the rules rather than having them pre-identified by humans. The intent of machine learning is to “generalize beyond the examples of the training set,” applying what we find to other/new instances.[2] For our purposes, we will use machine learning to provide a prediction of outcomes for new (or untrained) data, i.e. does a new transaction involve a nuclear-related strategic good or not. Two methods will be introduced in the following sections:

a) Supervised classification that creates prediction models based on licensed and unlicensed transaction data; and
b) Natural language processing with fuzzy matching, which compares text descriptions to flag transactions.

This paper is intended to promote further thinking and experimentation on applying machine learning in support of STC enforcement and outreach. It is only an introduction to the many ways these techniques can be used to detect trade in nuclear-related strategic goods.

SUPERVISED CLASSIFICATION WITH RESAMPLING

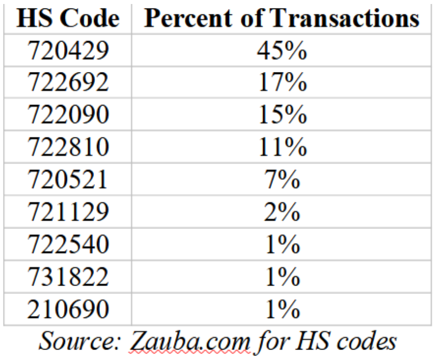
The first machine learning approach utilizes data on export licensed and unlicensed international trade transactions to create predictive classification models for nuclear-related strategic goods. It is a supervised learning approach, which means that models are created using a labeled training set and then applied to new instances to provide a prediction. In this case, the labels are the presence of an ECCN in trade declarations collected by state authorities. The classification model will take transactions, each labeled as either having a nuclear-related ECCN or not, and train on the different attributes of the transaction, such as HS code, destination, mode of transport, weight, quantity, value, unit value, etc. The model will then classify whether or not a transaction contains a nuclear-related strategic good. The model’s performance can be assessed using this training data and then applied to new transactions as they come in.[3] The following section walks through the steps in this process.

1) Data collection and preparation

The first step is to identify the nuclear-related strategic good(s) to create models for, based on their ECCNs. After the ECCN is identified, data for licensed export transactions that identify this ECCN in the shippers export declarations would be pulled together for a selected time frame. This time frame could be decided based on the amount of data, periodic reviews, or relevant events, like regulatory changes or geopolitical shifts. Once this data is gathered, we would create 'basket' of the different combinations of the ECCN and HS codes. This basket would show how often a particular HS code is utilized by exporters for transactions involving the particular ECCN (e.g. 45% of transactions involving strategic goods classified under ECCN X were shipped using HS code Y). This allows us to identify the HS codes that are *actively being used* for transactions involving a strategic good rather than relying on a static correlation table that says what the HS code for a strategic good *should be*.

An example of a such a basket for maraging steel (ECCN 1C116) is in Table 1. Note that the percentages are examples only and not based on actual transaction data.

Table 1. Example HS basket for maraging steel (ECCN 1C116)



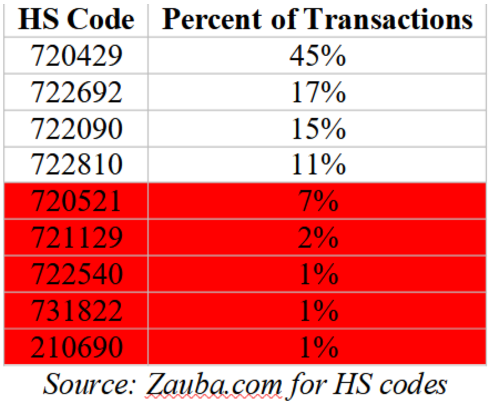
HS Code	Percent of Transactions
720429	45%
722692	17%
722090	15%
722810	11%
720521	7%
721129	2%
722540	1%
731822	1%
210690	1%

Source: Zauba.com for HS codes

The ECCN–HS correlations often contain codes that are utilized at a low rate. This may be due to misclassification, outlier cases, or other issues. In subsequent steps, transactions of commodities not subject to export licensing requirements for these HS codes will be gathered. As such, a reasonable cut-off point should be set so as to not include a large amount of transactions that may be irrelevant just because the HS code is in the basket. For example, a cut-off could be set at 10 percent—only HS codes utilised in 10 per cent or more of the transactions with an ECCN would be included in further analysis.

Table 2 shows the revised example basket for maraging steel with a 10 per cent cut-off.

Table 2. Example HS basket for maraging steel with cut-off



HS Code	Percent of Transactions
720429	45%
722692	17%
722090	15%
722810	11%
720521	7%
721129	2%
722540	1%
731822	1%
210690	1%

Source: Zauba.com for HS codes

Once the most used HS codes are identified, data is pulled from all transactions with the HS code but with no identified ECCN for the same time period from which we drew the transactions with ECCNs. This assembles the universe from which we can model the characteristics of trade in the particular strategic good. These steps produce a labeled dataset for supervised learning involving strategic goods and what we assume are non-strategic goods. Since we know the label of each transaction, we can test how well our models perform in predicting classifications.

2) SMOTE Resampling

In gathering the set of transactions identified above, transactions without an ECCN will greatly outnumber those that have an ECCN for the targeted commodity. In machine learning, imbalanced data can have adverse effects on modeling. For example, if we assume that every transaction does not involve a strategic good there may be a very high accuracy rate, but we will miss our objective: to detect the minority class or transactions involving nuclear-related strategic goods. In machine learning, "without considering the problem of class imbalance, the performance of learning algorithms is dominated by the majority class samples and the minority class samples are ignored as noise."[4] Since we are more interested in correctly classifying this minority class than the majority, we will resample our data to improve model performance.

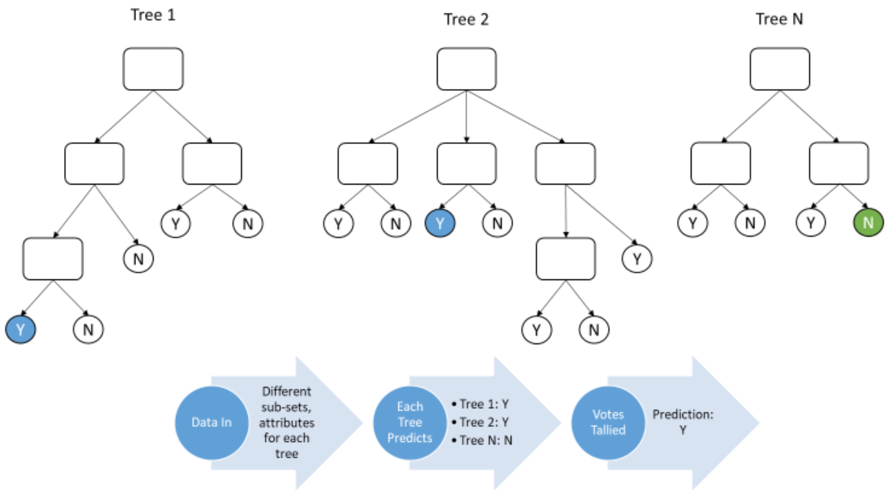
Techniques for resampling involve undersampling the majority class or oversampling the minority class. Undersampling involves removing transactions without ECCNs through random selection to bring balance with the number of transactions involving strategic goods.[5] Eliminating transactions could omit valuable information that might be important to the model. Oversampling involves duplicating transactions involving strategic goods to match the amount of transactions without an ECCN. This could create an overemphasis on certain characteristics, overfitting the model to these repeated transactions and creating a model that might not effectively predict new cases. Researchers have extensively examined the undersampling/oversampling trade-offs and have developed methods that "try to maintain structures or groups and/or generate new data according to underlying distributions."[6]

One such technique that has shown impressive performance in handling imbalanced data is the Synthetic Minority Oversampling Technique (SMOTE). This approach identifies similar examples in the minority class and creates new instances that share the space between them. Rather than duplicating transactions to oversample, this technique provides "new" examples of the minority class. It is important to note that resampling, through SMOTE or other methods, does not necessarily need to bring the majority and minority classes into exact balance, just to put them *more* in balance (e.g. 70/30, 60/40, etc.).

3) Modeling with Random Forest

Once prepared, the data is ready to be used to create a model. A minority portion of the collected dataset should be set aside for unbiased testing of the model once it is trained on the data, which will provide a measure of performance. This methodology proposes the use of the Random Forest algorithm to predict whether a transaction involves a strategic good. Originally proposed by Leo Breiman (2001), this model creates many decision trees based on randomly selected features and data samples to determine the classification of a transaction. A decision tree tests one feature at each decision node, splitting into sub-nodes in order to maximize the homogeneity of the resulting groups. It continues splitting until it can put the data, with a high probability, into a leaf that identifies the classification of the record. Random Forest expands this model by creating hundreds or thousands of randomly generated trees using different features and subsets of data. In our case, each tree would generate a prediction for each transaction and the final classification would be decided by a majority vote. Figure 1 shows a simplified representation of the Random Forest algorithm. Our models would be a binary classification; either the transaction is classified as involving a strategic good or not. In Figure X, two trees predict that the transaction does involve a strategic good and one does not, therefore the overall prediction is that there is a strategic good involved.

Figure 1. Random forest algorithm



This algorithm has many advantages. By utilizing random samples of features and data over many trees, it can help prevent overfitting the model. Individual decision trees, unless pruned or controlled, can continue splitting the dataset in increasingly specific ways creating a very specific model that cannot generalize to new data. In addition, Random Forest is very versatile, handling categorical and numeric data. Many machine learning algorithms require workarounds for categorical data. Applications of this algorithm also provide an ability to identify the importance of individual features to the final classification. This would allow the identification of which aspects of transactions, such as destination, HS code, value, etc., are most valuable in predicting whether a transaction involves a strategic good. By utilizing a multitude of trees, no individual test of the data will predominate, increasing confidence in the ultimate classification. Random Forest can be computationally expensive, however, so parameters of the algorithm may need to be adjusted for the amount of data involved.

The algorithm would be trained on the data and performance would be measured against the reserved test set, which has a label. Based on the test, parameters or features could be changed to increase performance. This model would apply to a particular strategic good or ECCN. After the approach has been tested, it can be used iteratively to create models for a broad portfolio of nuclear-related strategic goods. Once these models are created, they can be applied as new data arrives. The creation of multiple models has the advantage of taking into account of the characteristics of particular commodities in addition to providing a prediction of which specific product is involved.

Applying supervised classification to new transactions is a powerful tool to more effectively target resource intensive investigations or inspections. If a transaction is predicted to contain a strategic good and it has not been licensed, it could be a high priority for review. This method can also identify patterns of common misclassification or illicit trade in nuclear-related commodities. The models we design will recognize prominent behaviors, whether they are the use of particular HS codes for a strategic good, trade routes, or commodity valuations. If these patterns appear to be unusual, but are linked to specific strategic goods through modeling, we may have uncovered a proliferation pathway for further analysis.

NATURAL LANGUAGE PROCESSING

Natural Language Processing (NLP) is an umbrella term for a wide-range of techniques to gain knowledge from text data using machine learning. NLP has an expansive set of of applications, such as identifying writing patterns of different authors, conducting sentiment analysis, and performing smarter matching/searching (e.g. recognizing that “United States versus Canada” and “Canada vs. United States” is topically the same or matching “XYZ Corporation,” “X.Y.Z. Corp,” and “XYZ Corp Ltd.” as the same entity). There are also techniques to computationally extract key information from text, create automated translation systems, and perform text-based classification.

In international trade data, text comes primarily in the form of commodity descriptions provided in transaction-level documentation. These descriptions run the gamut in usefulness and specificity, with very few if any requirements that ensure uniformity. The same product, strategic good or otherwise, could be described in countless different ways. This mix, combined with the large volume of transactions, make it difficult to use commodity descriptions as a reliable information source for identifying nuclear-related strategic goods. There is a lot of potentially useful data there, but we need a way to meaningfully analyze it. NLP brings computing power to bear on the problem. In addition to the data from transactions, text in STC lists and other reference documents can be used as key references to match to commodity descriptions after being prepared with NLP techniques. This approach first utilizes algorithms to pre-process the text in commodity descriptions and reference documents. Once prepared, the commodity descriptions are assessed to find the degree to which they match the reference document descriptions for nuclear-related strategic goods.

1) Text Pre-Processing

Text pre-processing consists of a series of steps that are common in most applications of NLP. We will create a generalized pipeline of steps to prepare our text data in a way that is best suited for analysis. The following procedures are almost universal in NLP:

- Initial Cleaning and Normalization– The removal of unnecessary text, special characters, and expansion of contractions. This also involves conversion of all text to a standard format, usually making everything lowercase.
- Tokenization – The process of separating text into pieces and placing it into a usable data structure. Tokenization can be done by sentence or by word; for our purposes we will stick to word based tokenization. Punctuation is treated as a separate token. For example, "Tokenize this sentence for analysis." would be tokenized by word to a list of the following entries: ['Tokenize', 'this', 'sentence', 'for', 'analysis', '.']. Because tokenization separates on punctuation, contractions can cause some problems if they are not expanded beforehand.
- Lemmatization - Lemmatization "map[s] different variants of a word to its root. With this approach, the non-trivial inflections such as is, are, was, were, are mapped back to the root 'be.'"[7] Lemmatization is based on an understanding of the language to establish the true root of a word and relies on a lookup corpus to function.
- Removing Stop Words – Stop words are the most common words in speech and do not add value to the meaning of the text. They are high-frequency, low-content words, such as an, the, but, how, him, she, etc. Since these words do not add value, we do not want to use them in analysis of the text.

Text pre-processing allows us to "operationalize" strategic goods references, or create reference tables for strategic goods that we can use for matching against transaction records. These reference documents can be created from any text-based source that provides details and descriptions of nuclear-related strategic goods. In this case, one of the fundamental references is the Nuclear Suppliers Group lists. There are other relevant sources for nuclear commodity descriptions and keywords, such as the International Atomic Energy Agency's Physical Model and the United States's Alphabetical Index to the Commerce Control List. We can also utilize commodity descriptions from transactions that we have verified contain strategic goods. Combining all of these sources can create a dynamic and evolving reference list, with search terms that match regulatory documents, technical parameters, and real-world usages, like brand names.

As an example, the Alphabetical Index to the Commerce Control List, which links specific product types to ECCNs, was pre-processed.[8] This document, created by the United States government and freely available online, provides a searchable list of products and their associated ECCNs. For the purpose of this exercise, we identified the first ECCN as the primary ECCN associated with each description, which usually has a higher level of control than subsequent listed ECCNs. The Alphabetical Index was put through the pre-processing pipeline where it was cleaned and normalized, tokenized, lemmatized, and stop words were removed. The resulting dataset provides meaningful words associated with strategic goods per ECCN. Below is an example of the results for ECCN 3A233, which controls mass spectrometers.

Table 3. Key Words for ECCN 3A233

mass	9	thermal	2
spectrometer	9	glow	2
source	2	icp/ms	2
inductively	2	microfluorination	1
tims	2	plant	1
coupled	2	electron	1
plasma	2	beam	1
gdms	2	bombardment	1
ion	2	enrichment	1
ionization	2	molecular	1
discharge	2	uf6	1

When constructed, these references should connect a specific commodity label, such as the ECCN, to the set of processed keywords. This allows us to connect the similarity ratio discussed in the next section back to the nuclear-related strategic good. The frequency of words in this table should be used carefully. Frequency does not always impart increased significance. There are also some words that, if used as search terms, may not add value to the detection of strategic goods, such as "source." We may decide to remove words like these by creating a subject specific version of our stop words list to remove common terms like "equipment," "software," or "source."

2) Fuzzy Matching

Matching text is simple in concept, but can be quite difficult to perform in real-world situations. In this case, we want to use matching to check commodity descriptions from transaction records against the reference list of commodity descriptions constructed in the previous step. The problem lies in the wide-range of potential spellings, punctuation differences, word orderings and other uniqueness that will be present within the huge volume of commodity description data. Attempting to match on a 1-to-1 basis between our references and text from transaction records will not yield very good results. Some of these issues are resolved through text pre-processing, such as punctuation and getting the roots of words, but exact text-to-text matching from different sources is still a challenge.

Fuzzy matching is a more expansive form of string matching that uses algorithms to assess the similarity between the content of strings rather than trying to match them exactly. There are many different approaches to calculating string similarity. One such method is Levenshtein distance, which has been implemented with success in many data science and machine learning applications.[9] The algorithm for Levenshtein distance measures the difference between two strings as the number of transformations required to change, through insertion, deletion, or substitution, one string to match the other. For example, if the two strings we are comparing are "redacted" and "Reduction," the changes are

1. r → R
2. a → u
3. e → i
4. d → o

5. Add n

for a Levenshtein distance of 5. Note that changing a character case or adding/removing punctuation all increase the distance. These differences would be handled in text pre-processing so that we only measure this distance based on substantive changes in the text. Based on this calculation, the smaller the distance between strings, the closer the match. Levenshtein distance is often presented as a ratio between 0 and 1, with 1 being an exact match.

In Python, the FuzzyWuzzy package has a straightforward implementation of the Levenshtein distance calculation and multiple ways of calculating the resulting similarity ratio between strings.[10] Depending on analysts' preferences, the similarity ratio can be calculated on the texts exactly, ignoring punctuation and case, or ignoring word order within the string. Below are the steps for applying fuzzy matching for the identification of nuclear-related strategic goods transactions:

1. Gather transaction-level data and pre-process the text for commodity descriptions.
2. For each transaction, fuzzy match the commodity description against a constructed nuclear-related reference table. The fuzzy match calculates a similarity ratio based on the Levenshtein distance.
3. Return the strategic good reference entry for the highest similarity ratio for each commodity description. The entry returned will be the most likely match between the transaction commodity description and strategic good description. The similarity ratio will provide an indication of the confidence level in the match.
4. Based on study and testing, set a minimum viable similarity ratio for the matches. Transactions with similarity ratios above this value will be flagged as matches to the specific strategic good.[11]

The resulting match flags should be prioritized for further review. This process gives us a resource efficient way of processing a very large volume of text data from international trade transactions to check for strategic good description matches.

Fuzzy matching depends upon our confidence in the processed reference documents that we use for matching to strategic goods. Using STCs lists is a good starting point as they include keywords and parameters linked directly to specific nuclear-related goods. They are not, however, the only available resource that can be used as a reference. For example, STC lists do not include common trade names, brands, or other industry specific identifiers that might help increase our identification rate. Certain metals have trade designations that can help differentiate strategic goods from non-strategic metals, such as the 2000 or 7000 series for aluminum alloys. Through research and subject-matter expertise, terms like these can be added to reference tables to improve our matching efforts. If made into an ongoing project, the efficiency of commodity description matching will continue to increase over time.

CONCLUSIONS

The detection of illicit trade in nuclear-related strategic goods is a major challenge, but the data-rich environment of international trade lends itself to machine learning solutions. Many other fields have embraced machine learning to recognize complex patterns in large datasets and strategic trade should be no exception. The approaches introduced in this paper are general rubrics and should be adapted and refined based on understanding of the real-world data and experimentation. Both supervised classification and fuzzy matching can add value by evaluating and prioritizing transactions based on risk, promoting more effective direction of limited resources for intensive investigations. They can also be used with historical transaction data to identify parties to transactions that may require targeted outreach explaining their export control licensing obligations. This introduction merely scratches the surface of the potential uses of machine learning for detecting illicit trade in nuclear-related strategic goods. Moving forward, stakeholders in this area should continue to examine how to more effectively leverage existing data to promote international security and non-proliferation through strategic trade control enforcement and outreach.

REFERENCES

1 The Trigger List and Dual-use List (INFCIRC/254 Parts I and II) closely relate to the International Atomic Energy Agency's INFCIRC/540 Annex I and II. There have been changes over the years to key the former lists "up-to-date."

2 Domingos, Pedro, "A Few Useful Things to Know About Machine Learning," *Communications of the ACM*, no. 55:10 (2012),

3 Using ECCNs recorded on shipping documentation as labels are a circumstance where we use the best information available. In most cases, these labels capture properly licensed strategic goods transactions. They do not capture cases where the goods are intentionally or unintentionally misclassified, unlicensed, or outright smuggled. In part, what we hope is that by training a model based on known strategic goods transactions, we can uncover these illicit behaviors as new transactions come in that match the characteristics of a commodity that should be licensed.

4 Nwe, Mar Mar and Khin Thidar Lynn, "Effective Resampling Approach for Skewed Distribution on Imbalanced Data Set," *IAENG International Journal of Computer Science*, no. 47:2 (2020): 1.

5 In this case, undersampling is effectively the same as random sampling of the majority class. It creates a subset through random selection, which potentially omits information. Depending on the scale of our analysis and processing power, it may be necessary to perform this step as well. Analysts will have to consider their individual circumstances.

6 Krawczyk, Bartosz, "Learning from imbalanced data: open challenges and future directions," *Progress in Artificial Intelligence*, no. 5 (2016): 2.

7 Kasaraneni, Chaitanya Krishna, "Understanding NLP Pipeline," *Analytics Vidhya* (2020), https://medium.com/analytics-vidhya/understanding-nlp-pipeline-9af8cba78a56 (accessed 17 April 2021).

8 This text pre-processing was performed using Python and the NLTK natural language processing package.

9 There are many other algorithms to calculate string and word similarity, each with benefits and drawbacks. These are beyond the scope of this paper. For some other algorithms of interest see n-gram distance approaches, the Jaro-Winkler algorithm, Hamming distance, and adjustments to Levenshtein.

10 See Arias, Francisco Javier Carrera, "Fuzzy String Matching in Python," *Datacamp* (2019), https://www.datacamp.com/community/tutorials/fuzzy-string-python (accessed 19 April 2021).

11 FuzzyWuzzy outputs a similarity score from 0 to 100 based on the selected ratio.