Carnegie Mellon University
Software Engineering Institute

PRACTICAL SUPERVISED MACHINE LEARNING CLASSIFICATION OF HIGHLY IMBALANCED TEXT

Austin Whisnant

May 2025

DOI: 10.1184/R1/29120552

Introduction

Insiders are responsible for some of the most damaging cases of fraud, sabotage, and espionage in history. These cases have led to serious consequences in national security, weakened competitive advantages of private corporations, and harmed customers, employees, and citizens of governments and organizations around the world. Insider incidents occur across the public and private sectors, in a wide variety of industries, and over a range of organization sizes [1]. Reports of such incidents have increased substantially over the last decade [2], and insiders accounted for 19% of all breaches in 2023 [1].

As the threat grows and the problem becomes more widely understood, software vendors have started offering more solutions for detecting, preventing, and evaluating the risks of insiders. It is important that these and future solutions are founded on reliable data and evidence-based research. To that end, our objective with the research in this paper is to efficiently collect and classify United States Attorneys' Office (USAO) press releases to determine which ones describe an insider threat. The goal of doing this is to create an automated process to collect as many court cases as possible in which insiders have been prosecuted. Our efforts outside the scope of this paper will be to encode detailed data from these court cases to support future insider threat research.

Manually reading and labeling each press release as "insider threat" or not would be highly inefficient over the long term considering that there are around 50 press releases published per day. Using a keyword search for words such as "employee," "embezzle," or "fraud" is helpful but not particularly accurate. (See the Results section for these metrics.) Given the goal of collecting as many cases as possible as efficiently as possible, using machine learning methods is likely the best approach. The next section describes related work in both the insider threat and text classification domains. The Press Release Data section describes the USAO data, how it was collected, and the data labeling process. The section on Methods describes the implementation of the classifiers that were tested and is followed by the Results of those tests. Lastly, we conclude with a description of how we will apply the results in practice to further advance insider threat research.

Related Work

This work combines research from two separate domains: insider threat and machine learning. First, we summarize relevant work in the insider threat domain and then discuss previous machine learning approaches to text classification.

Insider Threat

According to the CERT Division of the Software Engineering Institute (SEI), an insider threat is defined as “the potential for an individual who has or had authorized access to an organization’s critical assets to use their access, either maliciously or unintentionally, to act in a way that could negatively affect the organization” [3]. Though it is a niche area of research, there is an extensive body of work related to detecting and preventing insider threats on organizations’ networks. CERT’s *Common Sense Guide to Mitigating Insider Threat* outlines 22 best practices for preventing insider threats, including separation of duties, knowing the organization’s critical assets, and learning from past incidents [4]. A large portion of insider threat literature focuses on behavioral analysis, while the rest focuses on monitoring the use of information technology (IT) for anomalies. For example, Schoenherr et al. argue that behavioral and social sciences offer better methods for detecting and predicting insider actions than IT methods do because of multiple issues that limit empirical research in IT detection and prediction [5].

There are also dozens of recent papers investigating new machine learning techniques for detection and risk scoring (i.e., prediction) of potential insiders [6] [7] [8]. These papers focus on detecting insider actions on organizations’ networks (in data sources such as file access logs, email, web logs, and so on) but not, as in our case, on classifying text documents that describe criminal actions. They evaluate supervised and unsupervised methods as well as neural network techniques. Many of these show promising results by evaluating anomalous behavioral activity on IT systems. However, the datasets on which many of these algorithms are based are limited [9].

We hope that the additional data provided through the detailed study of court cases identified by our algorithm will support future empirical research in both behavioral and technical methods.

Text Classification with Machine Learning

Researchers and analysts from across many disciplines are increasingly interested in using machine learning to classify text. Applications range from summarizing large amounts of text [10] to sentiment analysis [11], to fake news detection [12], and more. There are many papers that compare models, explain their differences, and summarize their pros and cons. Tzimourtas et al. compared support vector machines (SVMs), random forest, and naïve bayes using Term Frequency Inverse Document Frequency (TF-IDF) vectorization over a news group dataset; the result was that SVM had the highest accuracy [13]. Suneera et al. use the same news group dataset but included three more supervised techniques, five deep learning techniques, and three different vectorizers [14]. Their experiment showed that logistic regression with TF-IDF had the highest accuracy among all models and methods tested.

No baselines exist specifically for classifying USAO press releases, though some analyses have been conducted on other types of news articles. For example, Rahman experiments with classifying Bangla-language news articles using a grid search across 15 different algorithms, both supervised and unsupervised, and five different text vectorizers [15]. Their best results came from a bi-directional Long Short Term Memory (LSTM) model with a skip gram vectorizer, and the best supervised technique was from an SVM with a TF-IDF vectorizer. They also showed that classification via article text works slightly better than classification via article title. Bounabi et al. compare several supervised machine learning algorithms using Fuzzy TF-IDF vectorization to classify a set of BBC sports data. The results were excellent across all implementations, with Naïve Bayes having a slight edge over the others by AUC [16]. Shah et al. compared logistic regression, random forest, and KNN models with a TF-IDF vectorizer for classifying BBC new articles [17]. In their experiments, logistic regression has the highest accuracy and precision.

Press Release Data

We chose to use USAO press releases because they are free, easily accessible, semi-structured (partially categorical and partially free text), and correspond to federal court cases. By concentrating on data from court cases, future insider threat research can be based on data that has been validated by law enforcement and the legal system as opposed to relying on self-reported cases, surveys, or otherwise unvalidated, inconsistent data from news articles. Additionally, USAO press releases and their associated court records tend to be more consistently accessible than U.S. state cases or international cases.

The USAO publishes press releases for major updates to ongoing cases, such as when an individual is arrested, charged, or sentenced, as well as for general actions, announcements, and updates from the USAO. A typical press release will have a title, date, one or more “components” (e.g., “criminal division,” “tax division,” “U.S. Marshalls Service,”) or the USAO region (e.g., “USAO – West Virginia, Northern”) and a body of text consisting of 5-10 short paragraphs that describe the crime and often, the U.S. attorney’s reaction to it. Articles sometimes also have an associated topic, such as “tax” or “cybercrime.” There were 50 distinct topics in our dataset.

Data Collection

The Department of Justice offers an API for accessing press releases [18]. However, due to limitations with date range selection and our need to process articles on an ongoing basis, the API was not an effective option for our use. We therefore used standard web scraping tools, including Python and the Beautiful Soup library, to scrape the HTML from the USAO press release website. Even though the API did not support our goals, we followed its guidelines for developers regarding page and rate limits when conducting the web scraping. We limited our requests to one every five seconds and ran our scraper only in the middle of the night (EST) and only for the most recent articles. The press releases were scraped into a database with the following fields: title, url, date, district, topic(s), component(s), and id (auto-generated integer). The text of each article was saved in a text file titled with its corresponding id.

Table 1: Example USAO Press Release Entry

Field	Example Entry
title	Blackstone man arrested for defrauding former employer
url	[base url]/usao-ma/pr/blackstone-man-arrested-defrauding-former-employer
date	2023-12-06
district	District of Massachusetts
topic(s)	Financial Fraud
components(s)	USAO – Massachusetts
id	113
file	2023/12/113.txt

Data Labeling

For our base dataset, we manually labeled 8,986 press releases from September 1, 2023 through February 29, 2024. Some press releases could easily be classified by looking at their title. For example, "Project Safe Neighborhoods Logo Contest" is obviously not an insider threat article, but "Man Embezzles Money from Employer" clearly is. However, many articles could not be labeled by reading their title alone (e.g., "Man Sentenced for Federal Fraud Crime"), so the text of the article was opened and read before being manually labeled.

We used CERT's definition of insider threat to label the press releases [4]. However, the CERT definition is not always sufficient to determine whether a particular press release qualified as insider threat. For the purpose of coding consistency, we therefore developed more specific guidelines, shown in Table 2, for labeling common cases that needed extra guidance. For these guidelines, we focused on CERT's three necessary requirements: (1) use of authorized access to commit the crime, (2) the insider being part of a larger organization, and (3) harm, or potential harm, to the organization resulting from the incident.

Table 2: Labeling Guidelines for Common Edge Cases

Job Role	Crime	Label	Reasoning
Teacher/Coach	Child exploitation	Not InT	The teacher does not necessarily commit the crime against a child under their care as part of their employment. However, if, for example, a coach committed a crime against one of their players would be labeled an insider threat.
Police/Correctional Officer	Abuse of arrestee/inmate	InT	The insider violates organizational policies and leverages access to commit the crime.
Any	Kickback scheme	InT	The insider works for the victim of the scheme.
Vendor/contractor	Lying about qualifications	InT	Vendors and contractors are considered insiders.

Job Role	Crime	Label	Reasoning
Any	Insider trading	InT	Insider trading has the potential to cause reputational damage and possible legal liability.
Owner/Sole Proprietor	Any crime against customers/clients	Not InT	An example is a self-employed home care worker stealing from a client.

Some press releases describe bribery or kickback schemes in which the main subject of the article (the defendant) is an outsider but is aided in their scheme by an insider (who is often being prosecuted in a separate case). For articles about where the company policy was to commit the crime (e.g., an individual being prosecuted for illegally dumping hazardous waste for a company whose policy it was to do so), we did not label the press release as insider threat.

In cases where the label was still unclear after reading the article, we chose to err on the side of true (i.e., labeling the article as insider threat). For example, if an article stated that a police officer sold drugs, and it hinted but did not explicitly state that the officer used their job to obtain the drugs, we labeled it as insider threat. This allows us to pull more detailed court case information later in the process and make a final determination as to whether the case is an insider threat case after reviewing the court documents. The USAO article label remains the same, whether the case is later determined to be insider threat or not.

Statistics of Collected Data

The USAO released 8,986 articles between September 1, 2023 and February 29, 2024. We manually classified 6.4% (571) as insider threat articles and the rest as not insider threat. Table 3 compares statistics between the two classifications.

Table 3: Statistics of Collected Data

	Insider Threat	Not Insider Threat
Top 3 Components	FBI, Civil Rights – Criminal Section, Civil Rights Division	FBI, ATF, DEA
Top 3 Topics	financial fraud, public corruption, civil rights	drug trafficking, firearms offenses, violent crime
Avg Char Length (title)	92 (standard deviation of 30)	85 (standard deviation of 29)
Avg # of words (text)	462 (standard deviation of 275)	425 (standard deviation of 257)
Common words (title)	former, fraud, officer, employee, scheme	man, years, trafficking, fraud, drug
Common words (text)	scheme, company, former, pay, employee	firearm, possession, violent, intent, vehicle

Methods

The overview of our methodology is shown in Figure 1. First, we collected the data as described in the previous section, then we manually labeled it according to CERT's definition of insider threat and our own guidelines for handling edge cases as defined in Table 2. We then pre-processed the data for use with a set of classifiers and implemented those classifiers with a selection of hyperparameters using a grid search for the best performing model. All work was done in Python with Scikit-Learn.

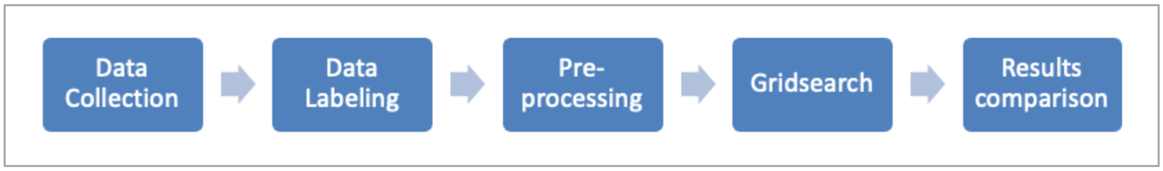

Figure 1: Classification Process

Pre-Processing

Prior to implementing the classifiers, we standardized the article text and title by converting all text to lower case, removing all punctuation, and encoding with utf-8. Of the eight available features in the data, we used four in the classifiers: title, topic(s), component(s), and the text of the article. We did not use the article id and date, as those should be irrelevant to whether a press release is insider threat or not. We also dropped the URL, as it duplicates the title field and provides no other relevant information. We dropped the district feature after confirming the lack of correlation between the district (location) and the classification with a Cramer's V test (resulting in an effect size of 0.17). We also removed all district level components (e.g., "USAO – Massachusetts") from the data, leaving 57 distinct (non-location) components.

Encoding

Due to the high number of categories in both the target and component features, we decided to test both one-hot encoding and target encoding to encode those features as numeric data. One-hot encoding is a commonly used technique to map categories to columns with binary values. However, it tends to increase dimensionality in the dataset, which can slow down the training of the model [19]. Target encoding tries to solve the problem of dimensionality by replacing the category labels with their prior probability of predicting the target variable (i.e., the likelihood that a particular category label predicts a true value for the target) [20].

Since the topic and component features in our dataset both have multilabel data (i.e., the topic of an article can have zero, one, or multiple topics), we used a multilabel binarizer to do the one-hot encoding. For target encoding, we calculated the predictive probability of each label (including a "none" label) and then chose the maximum probability for features with multiple labels. We chose to use max over mean or min because we wanted to ensure we collected as many true positives as possible. (The pseudocode for our multilabel target encoder implementation can be found in Appendix B.)

Vectorization

Prior to vectorization, titles and text were lemmatized using NLTK's WordNetLemmatizer. Titles and text were then vectorized into unigrams as separate features using a TF-IDF vectorizer. We chose TF-IDF because, as our literature review shows, it consistently performs extremely well for supervised learning methods. We used TF-IDF with the "english" stop words set provided with Scikit Learn, a minimum document frequency of 10, and a maximum document frequency of 70%.

Classifier Implementation

We chose to limit our classifier selection to traditional supervised machine learning methods because they are generally fast, easy to implement, cheap to run, and explainable, and because experiments from similar domains have achieved excellent results with supervised learning. We implemented a grid search across seven different classifiers and used recall as the metric for choosing the best classifier and hyperparameters. We used stratified 10-fold cross validation for testing, and we refitted the model after choosing the best parameters so that each classifier could be compared against the others using their best performing hyperparameters. (The full list of hyperparameters tested can be found in the Appendix.)

Results

Our key metrics are recall and false positive rate (FPR). Accuracy is not a useful metric for this dataset for two reasons: (1) it is a highly imbalanced dataset (only 6.4% of press releases are insider threat), and (2) we are sensitive to false negatives (i.e., we do not want to miss an insider threat case). False positives are acceptable if the FPR, the ratio of false positives to actual negatives, is not overly burdensome to the individuals that will read through the identified insider threat cases (i.e., it does not produce too many incorrectly classified cases per day for a human to reasonably be expected to read through).

For the sake of completeness, we also include the commonly used metrics of precision, accuracy, f1-score, and mean time to score, as well as a Kappa statistic to compare to the accuracy and help with understanding the models' performance on the highly imbalanced data.

Baseline

To understand whether a machine learning classifier is useful at all, we will compare the results of our classifier with that of a basic keyword search on the titles of the USAO press releases. We ran a frequency count of words in the insider threat press release titles versus the non-insider threat titles. We then used words that were common only to the insider threat press release titles, such as "former," “employ*,” “embezzle*,” “wire,” “bribery,” “insider,” and “contractor” to search across the sample dataset. The resulting confusion matrix is shown in Table 4.

Table 4: Confusion Matrix for Keyword Search

		Actual Values	
		Insider Threat	Not Insider Threat
Predicted Values	Insider Threat	395	452
	Not Insider Threat	176	7963

Therefore, the metrics for the keyword search are as follows.

Table 5: Keyword Metrics

	Recall	**FPR**	**Precision**	**Accuracy**	**F1**	**Kappa**	**Time**
Keyword Title Search	0.69	0.05	0.47	0.93	0.56	0.52	N/A

Classifier Results

As shown in Table 6, the model with the best recall using one-hot encoded categories was from the decision tree algorithm. However, the FPR for the decision tree model was a very high 0.757 and had an extremely low accuracy. The second-best performing model in terms of recall was random forest, which had an FRP of 0.216. This FRP is acceptable because, at an estimated 10 articles per day classified as insider threat, analysts would only need to read an extra two articles per day that result in false positives.

Table 6: Classifier Results for One-Hot Encoding

	Recall	**FPR**	**Precision**	**Accuracy**	**F1**	**Kappa**	**Time**
KNN	0.028	0.014	0.121	0.926	0.046	0.023	41.28
Random Forest	**0.902**	**0.216**	0.222	0.791	0.356	0.282	0.069
Gradient Boosting	0.500	0.007	0.837	0.960	0.626	0.607	0.053
Logistic Regression	0.837	0.114	0.345	0.883	0.488	0.435	0.033
Decision Tree	**0.982**	**0.757**	0.080	0.289	0.149	0.036	0.036
Multinomial NB	0.863	0.145	0.281	0.856	0.424	0.365	0.034

The target encoding method for the categorical features did not perform significantly better than the one-hot encoding method, and in some cases, it performed slightly worse.

Table 7: Classifier Results for Text Encoding

	Recall	**FPR**	**Precision**	**Accuracy**	**F1**	**Kappa**	**Time**
KNN	0.310	0.011	0.164	0.927	0.052	0.032	31.73
Random Forest	0.880	0.227	0.212	0.780	0.341	0.264	0.047
Gradient Boosting	0.526	0.007	0.836	0.963	0.646	0.627	0.054

	Recall	FPR	Precision	Accuracy	F1	Kappa	Time
Logistic Regression	0.836	0.150	0.279	0.849	0.418	0.355	0.041
Decision Tree	0.900	0.722	0.079	0.318	0.146	0.031	0.036
Multinomial NB	0.859	0.113	0.345	0.885	0.493	0.441	0.034

Class Weight Hyperparameter Performance

For the algorithms that included a 'class-weight' hyperparameter, the balanced mode always outperformed the unbalanced mode across all parameters searched. For example, the parameter options in the decision tree algorithm produced 40 different testing combinations. All the top 20 parameter combinations had a balanced class weight, with the best score being 0.98 and the worst being 0.52. The top performing unbalanced parameter combination had a score of 0.021, a difference of 83% from the worst performing balanced mode. This was consistent across the other algorithms offering class balancing, as shown in Table 8.

Table 8: Class Weight Performance

	Table Heading			
	Balanced		Unbalanced	
	Best	Worst	Best	Worst
Random Forest*	0.902	0.669	0.021	0
Logistic Regression	0.837	0.664	0.581	0
Decision Tree	0.982	0.515	0.212	0

* Balanced and balanced-subsample had the same best and worst recall scores for random forest.

Conclusions

To build our repository of insider threat court cases, we chose the random forest model as the most suitable model for our purposes. We have used this model to classify the entire corpus of over 200,000 USAO press releases going back to 2013 and will continue to use it going forward to collect new cases to include in our insider threat repository. Currently, 24,000 articles (12%) are classified as insider threat by the top model.

We plan to use the classifier on an ongoing basis to find new insider threat cases. To ensure the integrity of the data, we will manually classify press releases below a certain probability threshold as identified by the top classifier and randomly choose press releases to manually classify on an ongoing basis. We will also run a reduced version of the grid search once per quarter as more data is collected, and the classification labels are manually confirmed. This approach will allow us to monitor recall and false positive rates and to retrain the classifier(s) as the dataset grows. For future work, we plan to conduct a study of the press releases classified as insider threat to identify specific features of the cases that may be useful insider threat researchers.

We hope that the encoded details from the cases identified by our work will help support further research into preventing, detecting, and mitigating insider threat, and that our exploration of data classification techniques will inform research in other disciplines working with similar resources.

Bibliography

[1] Verizon, “2023 Data Breach Investigations Report,” Verizon, 2024.

[2] Securonix, “2024 Insider Threat Report: Trends, Challenges, and Solutions,” Cybersecurity Insiders, 2024.

[3] Software Engineering Institute, “Common Sense Guide to Mitigating Insider Threats, Seventh Edition,” Carnegie Mellon University, Software Engineering Institute’s Digital Library, 2022.

[4] CERT National Insider Threat Center, “Common Sense Guide to Mitigating Insider Threats Seventh Edition,” Carnegie Mellon University, Pittsburgh, PA, USA, 2022.

[5] J. R. Schoenherr and R. Thomson, “Insider Threat Detection: A Solution in Search of a Problem,” in *2020 International Conference on Cyber Security and Protection of Digital Services (Cyber Security)*, Dublin, Ireland, 2020.

[6] R. Nasir, M. Afzal, R. Latif and W. Iqbal, “Behavioral Based Insider Threat Detection Using Deep Learning,” *EEE Access,* vol. 9, pp. 143266-143274, 2021.

[7] D. C. Le, N. Zincir-Heywood and M. I. Heywood, “Analyzing Data Granularity Levels for Insider Threat Detection Using Machine Learning,” *IEEE Transactions on Network and Service Management,* vol. 17, no. 1, pp. 30-44, 2020.

[8] T. K. Rao, N. Darapaneni, A. R. Paduri, A. G. S, A. Kumar and G. Ps, “Insider Threat Detection: Using Classification Models,” *Proceedings of the 2023 Fifteenth International Conference on Contemporary Computing,* p. 307–312, 2023.

[9] S. Yuan and X. Wu, “Deep learning for insider threat detection: Review, challenges and opportunities,” *Computers and Security,* vol. 104, 2021.

[10] A. Mishra, A. Sahay, M. a. Pandey and S. S. Routaray, “News text Analysis using Text Summarization and Sentiment Analysis based on NLP,” in *2023 3rd International Conference on Smart Data Intelligence (ICSMDI)*, Trichy, India, 2023.

[11] C. G. Inovero, G. J. P. Ditablan, J. R. C. Reyes and R. E. Tajanlangit, “A Sentiment Analysis Classification of Product Reviews through Convolutional Neural Networks (CNN),” in *2022 10th International Conference on Information Technology: IoT and Smart City (ICIT ‘22). Association for Computing Machinery*, New York, NY, 2022.

[12] T. A. Roshinta, Hartatik, E. K. Fauziyah, I. F. Dinata, N. Firdaus and F. Y. A'la, "A Comparison of Text Classification Methods: Towards Fake News Detection for Indonesian Websites," in *2022 1st International Conference on Smart Technology, Applied Informatics, and Engineering (APICS)*, Surakarta, Indonesia, 2022.

[13] A. Tzimourtas, S. Bakalakos, P. Tselenti and A. Voulodimos, "An exploration on text classification using machine learning techniques," in *25th Pan-Hellenic Conference on Informatics (PCI '21). Association for Computing Machinery*, New York, NY, 2022.

[14] C. M. Suneera and J. Prakash, "Performance Analysis of Machine Learning and Deep Learning Models for Text Classification," in *2020 IEEE 17th India Council International Conference (INDICON)*, New Delhi, India, 2020.

[15] R. Rahman, "A Benchmark Study on Machine Learning Methods using Several Feature Extraction Techniques for News Genre Detection from Bangla News Articles & Titles," in *2020 7th International Conference on Networking, Systems and Security (NSysS)*, Dhaka, Bangladesh, 2020.

[16] M. Bounabi, K. E. Moutaouakil and K. Satori, "Text classification using Fuzzy TF-IDF and Machine Learning Models," in *4th International Conference on Big Data and Internet of Things (BDIoT '19). Association for Computing Machinery*, New York, NY, 2020.

[17] K. Shah, H. Patel, D. Sanghvi and M. Shah, "A Comparative Analysis of Logistic Regression, Random Forest and KNN Models for the Text Classification," *Augmented Human Research,* vol. 5, no. 12, 2020.

[18] Department of Justice, "API Documetation Version 1," https://www.justice.gov/developer/api-documentation/api_v1. [Online]. [Accessed March 2024].

[19] P. Cerda and G. Varoquaux, "Encoding High-Cardinality String Categorical Variables," *IEEE Transactions on Knowledge and Data Engineering,* vol. 34, no. 3, 2022.

[20] F. Pargent, F. Pfisterer, J. Thomas and B. Bischl, "Regularized target encoding outperforms traditional methods in supervised machine learning with high cardinality features," *Computational Statistics,* vol. 37, 2022.

[21] D. Mundie, S. Perl, C. Huth and C. Mellon, "Insider Threat Defined: Discovering the Prototypical Case," *J. Wirel. Mob. Networks Ubiquitous Comput. Dependable Appl.,* 2014.

[22] United States Attorney's Office, "Unauthorized Access Led to Deletion of 16,000 WebEx Teams Accounts in the Fall of 2018," Security Magazine, 26 August 2020. [Online]. Available: https://www.justice.gov/usao-ndca/pr/san-jose-man-pleads-guilty-damaging-cisco-s-network. [Accessed March 2024].

[23] V. Bergengruen and W. Hennigan, "The Strange Saga of Jack Teixeira Reveals New Security Challenges," 13 April 2023. [Online]. Available: https://time.com/6271787/jack-teixeira-arrest-leaks/. [Accessed March 2024].

[24] Federal Bureau of Investigation, "Former Boeing Engineer Sentenced to Nearly 16 Years in Prison for Stealing Aerospace Secrets for China," 8 February 2010. [Online]. Available: https://archives.fbi.gov/archives/losangeles/press-releases/2010/la020810.htm.

Appendix A: Results

Hyperparameters Tested

Logistic Regression

- Solvers: newton-cg, lbfgs, liblinear
- C: 0.1, 0.01, 0.001
- Class weight: None, Balanced
- Max iterations: 100, 1000

Random Forest

- Max depth: 3, 5, 10
- Max features: sqrt, log2
- Min samples split: 2, 5, 10
- N estimators: 100, 200, 500
- Class weight: balanced, balanced subsample, none

Decision Tree

- Criterion: gini, entropy
- Max depth: 1, 3, 5, 7, 9
- Max features: sqrt, log2
- Class weight: none, balanced

K Nearest Neighbors

- Weights: distance, uniform
- Metric: manhattan, euclidean
- N neighbors: 3, 5, 9, 35

Gradient Boosting

- N estimators: 100, 300, 1000
- Max features: sqrt, log2
- Min samples split: 0.1, 0.3, 2, 10
- Max depth: 3, 7, 11, 15

Multinomial Naïve Bayes

- Fit prior: True, False
- Alpha: 0.01, 0.1, 0.2, 0.5

Best Hyperparameters

The following are the best hyperparameters for each classifier using one-hot encoding:

Logistic regression: *Solvers:* newton-cg or liblinear, *C:* 0.01, *Class weight:* balanced

Random Forest: *Max depth:* sqrt, *Max features:* 3, *Min samples split:* 2, 5, 10**,** *N estimators:* 500, *class weight:* balanced, balanced subsample

Decision Tree: *Criterion*: gini/entropy, *max depth*: 1, *max features*: sqrt, *class weight*: balanced

K Nearest Neighbors: *Weights*: distance, *Metric*: Manhattan, *N neighbors*: 5

Gradient Boosting: *N estimators*: 1000, *Max features*: sqrt, *Min samples split*: 0.3, *Max depth*: 11

Multinomial Naïve Bayes: *Alpha*: 0.1, *fit prior*: False

Appendix B: Pseudocode

The following pseudocode describes an original technique for implementing target encoding on multilabel categorical data. This example uses the "topic" feature.

```
# Caculate probabilties for each topic in the list of all potential
# topics
target_probabilities = {}
for topic in all_topics:
  articles_with_topic = count(articles with this topic)
  # Calculate the likelihood for this particular topic
  target_probabilities[topic] = articles_with_topic / len(articles)

# Calculate the probability when no topic (i.e., empty feature)
articles_with_no_topic = count(articles with empty topic list)
no_targets_prob = articles_with_no_topic / len(articles)

# Reassign probabilities to the Topic feature, taking the max
for article in articles:
  if article has no topics:
    article topic = no_targets_prob
  else:
    article topic = max(topic probability of all topics in article)
```

Legal Markings

The view, opinions, and/or findings contained in this material are those of the author(s) and should not be construed as an official Government position, policy, or decision, unless designated by other documentation.

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN “AS-IS” BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

DM25-0718

Contact Us

Software Engineering Institute
4500 Fifth Avenue, Pittsburgh, PA 15213-2612

Phone: 412/268.5800 | 888.201.4479
Web: www.sei.cmu.edu
Email: info@sei.cmu.edu