Predicting Real-Life Disasters on Twitter Through Machine Learning

Samantha Lorenzo
EDLD654 Machine Learning for Educational Data Science
Final Project Report
December 6, 2023

Reproducibility Note: Please use this link to view the GitHub repopsitory. Instructions on how to maneuver throughout the repository can be found in the **README.md** file within the **Main Branch**. All of the coding used for this final project report can be found in the **r-code.r** file within the **R Branch** or through the **R Code** Kaggle notebook. All of the data files used for this final project report can be found within the **Data Branch** of the GitHub repository.

Introduction

Since the beginning of humankind, civilizations across the globe have accumulated a perpetual rolodex of disasterous events – from conflicts to environmental disasters to virus outbreaks (Roos, 2022). For instance, in 2022, the History Channel released an online article that outlined some of the most monumental crises from over the past century, aptly titled “The Deadliest Events in US History” (Roos). The list includes warfare (i.e., the U.S. Civil War, World War I, World War II, the Vietnam War, and the Korean War), acts of terrorism (i.e., 9/11), natural disasters (i.e., the 1900 Galvestone hurricane and the 1906 San Francisco earthquake and fire), and health crises (i.e., the 1918 flu pandemic, the HIV/AIDS epidemic, and the COVID-19 pandemic). Given the broad spectrum of categories and increasing advancements in technology, the ways in which modern communication channels can enhance the spread of information about disasterous occurrences are seemingly insurmountable. As current events continue to redefine societal norms across the globe, digital culture remains as the cornerstone for contemporary social interactions and information dissemination. Social media in particular has become a prime tool for discussing disasters in the form of digital narratives.

Background

Humans are instinctively drawn to narratives because story structures help organize experiences and document significant events. Stories enable people to live vicariously through narratives and consider the concepts being addressed. One popular form of narration is the act of micro-blogging, a mode of communication composed of user-generated posts which can be dispersed throughout

the internet via internal messaging features and other technological tools, such as smartphones, email, and online publications (Java et al., 2007).

Since its founding in 2006, Twitter (now X) has hailed as the premier microblogging source for its digital community and as of 2020, the platform's global network amassed more than 409 million users ("Forecast of the number of Twitter users," 2021). Over the past decade, content on Twitter has transformed and adapted in response to current events and matters of public interest, particularly those related to disasters around the globe and political and social affairs. Information consumption and dispersion continue to prevail as primary motivators for Twitter usage, especially among Americans (Mitchell et al., 2021), along with utilizing the platform as an online communication blog (or web log) for sharing and distributing personal points of view.

Research Problem

As mentioned, Twitter has become a key communication tool for important events, including disasters. The pervasiveness of the web-based smartphone application offers people a platform to announce an emergency they are witnessing in real-time. Due to this capability, more entities are increasingly becoming interested in monitoring Twitter to garner and provide life-saving information and keep disaster relief organizations and news agencies abreast of emergency situations. However, it is difficult to know for certain whether a tweet is actually announcing a disaster or if its context are rooted in something unrelated (e.g., a tweet with the word "ablaze" in the text could be meant metaphorically rather than to depict an active fire). So then, how can invested agencies get a better sense of when a tweet is referring to an actual disaster?

Through this report, my goal is to develop a machine learning model that predicts which tweets are about real-life disasters and which are not. Through data complied by Howard et al. (2019) and produced as a Natural Language Processing competition on Kaggle, I examined 10,000 hand-classified disaster tweets to create a series of predictive models and determine which is the best to mitigate the research problem at hand.

Description of Data

The overall data observed is comprised of three separate files: a sample submission, a test set, and a training set. Each sample in the training and test datasets encompass the *text* of a tweet, a *keyword* from each tweet, and each tweet's *location* of origin. In addition, the columns within the sets are organized as follows: **ID** (i.e., a unique identifier for each tweet), **Text** (i.e., the text of the tweet), **Location** (i.e., where in which the tweet was sent; some cells may be blank), **Keyword** (i.e., a particular word classifier for the tweet; some cells may be blank), and **Target** (i.e., whether a tweet is about a real disaster or not; included in the training dataset only). In reference to the Target category, tweets that are predicted to cite an actual disaster are coded as **Disaster** (which I recoded as **1**) while those that do not are coded as **Otherwise** (which I recoded as **0**).

I began processing the data by importing the competition's provided "Getting Started" folder into Kaggle, which then read the sample submission, test set, and training set files. To simplify the vast variety of tweet locations in the data, I created a new **country** column that indicates if a tweet was produced in the **United States**, **United Kingdom**, **Other**, or **Unsure**. Next, I used the **finalfit package** to view descriptive statistics of the data through both continuous and categorical tables. Through this, I was able to see that the data had no missing values.

$Continuous

A data.frame: 2 × 12

	label	var_type	n	missing_n	missing_percent	mean	sd	min	quartile_25	median	quartile_75	max
	<chr>	<chr>	<int>	<int>	<chr>	<chr>	<chr>	<chr>	<chr>	<chr>	<chr>	<chr>
id	id	<int>	7613	0	0.0	5441.9	3137.1	1.0	2734.0	5408.0	8146.0	10873.0
target	target	<int>	7613	0	0.0	0.4	0.5	0.0	0.0	0.0	1.0	1.0

$Categorical

A data.frame: 3 × 9

	label	var_type	n	missing_n	missing_percent	levels_n	levels	levels_count	levels_percent
	<chr>	<chr>	<int>	<int>	<chr>	<int>	<chr>	<chr>	<chr>
keyword	keyword	<chr>	7613	0	0.0	222	-	-	-
location	location	<chr>	7613	0	0.0	3342	-	-	-
text	text	<chr>	7613	0	0.0	7503	-	-	-

I then proceeded to load the **reticulate** and **sentence_transformer packages**. Using the **RoBERTa model** (available on Hugging Face) as my pre-trained language model, I was able to encode any text sequence up to 512 tokens as input and then return a numerical vector with a length of 768 that represent the text sequence. Next, I merged the embeddings with the main datasets, dropping the Text and Location columns since they were no longer needed after the embeddings were obtained. I then grouped the numeric and categorical variables together.

Applying the **recipe package**, I was able to declare my variables of interest to be used for modeling by assigning different roles (i.e., **id**, **predictor**, and **outcome**), as well as the processes to be applied. I then mixed the ingredients in the recipe by applying the **prep function** to the training data, which was then transformed into a new data set. Through this, the output object printed the number of variables for each assigned role: one outcome, 770 preductors, and one id. Then, in order to obtain a realistic measure of model performance, I split the data into two subsamples: training (**tweet_tr**; 6,090 tweets) and test (**tweet_te**; 1,523 tweets). Due to the relatively small sample size, I used an 80/20 split. The smaller test dataset will be used as a final hold-out set and the training dataset will be used to build my models. From here, I am able to use the **caret package** to build three predictive models.

Description of the Models

For this project, I decided to produce a **logistic regression model with no regularization**, a **logistic regresssion model with ridge penalty**, and a **logistic regression with lasso penalty**. Below are further details about each model, as well as the specific settings used for model fitting.

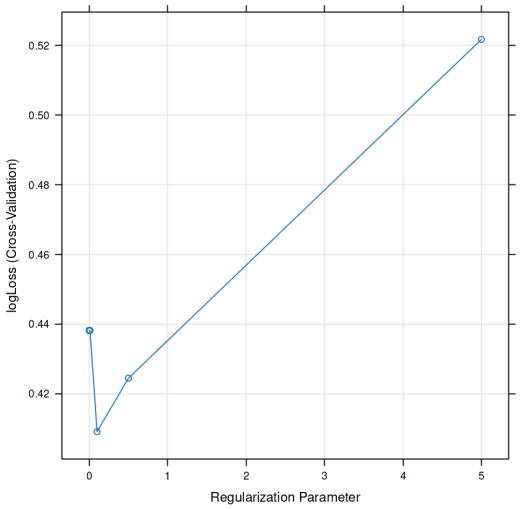
Model 1: Logistic Regression with No Regularization

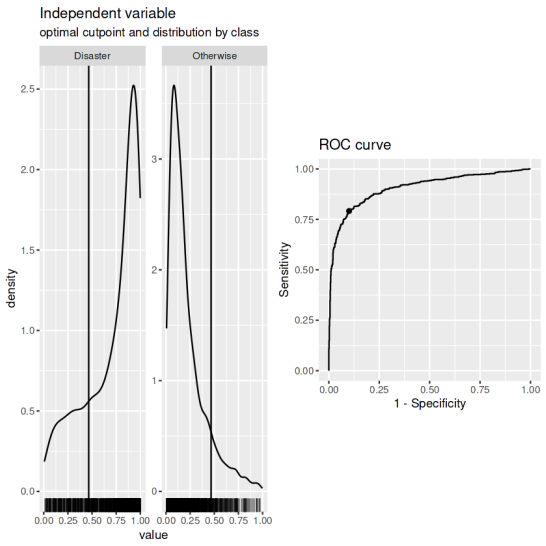
Logistic regression is a broadly used classification algorithm that applies a linear model to "predict the probability of a binary outcome" (RITHP, 2023). In this case, the binary outcome to be predicted is whether or not a tweet is referring to an actual disaster (i.e., "Disaster" or "Otherwise"). For this model, I used cross-validation to calculate the log loss, a evaluation metric for binary classification models that measures performance "by quantifying the difference between predicted probablities and actual values" (Setia, 2023).

Model 2: Logistic Regression with Ridge Penalty

Ridge regression is a convenient problem-solving option when there is "less than 100,000 samples or when there are more parameters than samples" (Maklin, 2018). As the data for this project falls under the former criterion, I thought it would be worthwhile to create this type of model. In addition, ridge (and lasso) penalties are used to "prevent overfitting in regression models by adding regularization terms to the cost function" (Jain, 2023). This stimulates the development of simpler models characterized by fewer predictors, enhancing their overall generalizability. For this model, I used a hyperparameter tuning grid for ridge penalty (lamba) with alpha being zero.

- **Data Visualizations**

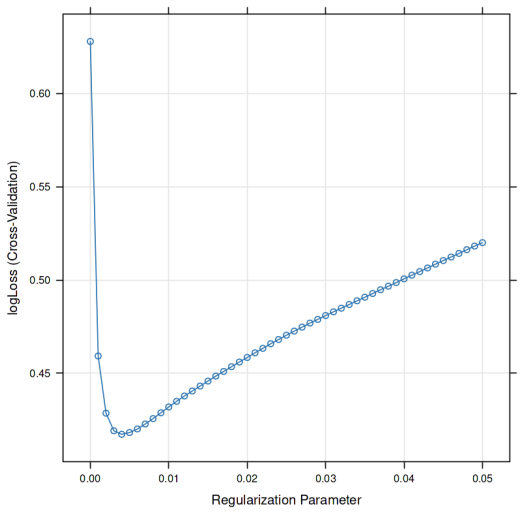


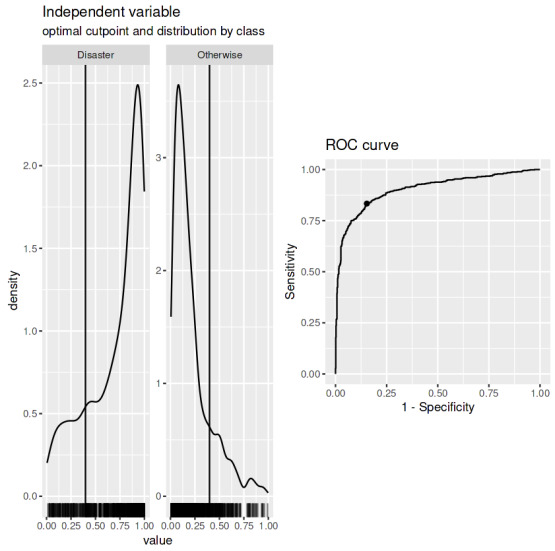


Model 3: Logistic Regression with Lasso Penalty

Lasso regression is a regularization technique (Jain, 2023). It is often preferred over other regression techniques to enhance prediction accuracy. This model incorporates a technique called *shrinkage*, which involves pulling data values closer to a central point (typically the mean). The lasso procedure specifically promotes the creation of simple and sparse models, characterized by having fewer parameters. Similar to Model 2, I used a hyperparameter tuning grid for lasso penalty (lambda). However, this time, alpha was set to one.

- **Data Visualizations**





Model Fit

In order to decide which predictive model is the best fit to solve the research problem, I evaluated their performance. I calculated and reported logLoss (Log Loss), the area under the reciever operating characteristic curver (AUC), overall accuracy, true positive rate (TPR), true negative rate (TNR), and precision for all three models. The numbers are summarized in the table below.

The calculations for Models 2 (Logistic Regression with Ridge Penalty) and 3 (Logistic Regression with Lasso Penalty) do not appear to have significant differences. For each respecitive model, AUC came out to 0.906 and 0.903, accuracy came out to 0.151 and 0.16, true positive rate came out to 0.159 and 0.165, true negative rate came out to 0.137 and 0.151, and precision came out to 0.137 and 0.151. In addition, both log loss results came out to 0.409. Conversely, while Model 1 (Logistic Regression with No Regularization) has higher log loss (0.671) than the other two models, it has the second highest precision (0.148; 0.003 behind Model 3) and the highest accuracy (0.186). Therefore, I believe Model 1 is the best out of the three to predict which tweets indicate a real-life disaster and which tweets do not.

A data.frame: 3 × 7

Model	Log Loss	AUC	Accuracy	TPR	TNR	Precision
<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
Logistic Regression	0.671	0.873	0.186	0.205	0.148	0.148
Ridge Logistic	0.409	0.906	0.151	0.159	0.137	0.137
Lasso Logisitic	0.409	0.903	0.160	0.165	0.151	0.151

Discussion

Given the pros mentioned for using ridge regression (e.g., it stimulates the development of simpler models characterized by fewer predictors) and lasso regression (e.g., it enhances prediction

accuracy through shrinkage), I did not expect the regluar logistic regression model to produce better results. Now that it has been determined that Model 1 is likely the best fit over its counterparts, I used the **coef function** to determine the 10 most important predictors of Target types (i.e., "Disaster" vs "Otherwise") and their regression coefficients. The results are displayed below. It seems that **V589**,**V218**,**V78**, and **V331** are the top predictors based on their larger absolute coefficients in the model.

```
     V589      V218       V78      V331       V83       V98      V495      V454 
22251.236 18149.671 16537.503 15388.949 10454.031  9213.427  8764.649  7747.010 
      V62      V750 
 7232.767  6775.940 
```

Conclusion

Overall, the work done for this final project has provided practical insights related to my research. Similar to the key purposes of the Kaggle competiton in which the data used was found, activist-oriented exploration and prosocial outcomes are the cornerstones of my research interests. My intent is to provide preventative measures for avoidable socio-psychological issues in efforts to promote a more cohesive global community for generations to come. My area of specialization lies at the intersection of persuasion and media psychology, technology and society, and media and public life. This focus inspires me to investigate tragic events (like disasters), social issues, and phenomena surrounding conspiracies and distrust. Through this project, I have garnered further insight into how content from digital communication tools can be processed and how predictive modeling can be used to cultivate effective solutions for avoidable communication problems.

References

(2021). Forecast of the number of Twitter users in the World from 2017 to 2025. *Statista*.

Howard, A., devrishi, Culliton, P., & Guo, Y. (2019). Natural Language Processing with Disaster Tweets. *Kaggle*. https://kaggle.com/competitions/nlp-getting-started.

Jain, A. (2023). Ridge and Lasso Regression in Python | Complete Tutorial (Updated 2023). *Analytics Vidhya*.

Java, A., Song, X., Finin, T., & Tseng, B. (2007). Why we Twitter: Understanding microblogging usage and communities. *Lecture Notes in Computer Science*, 5439.

Maklin, C. (2018). Machine Learning Algorithms Part 11: Ridge Regression, Lasso Regression And Elastic-Net Regression. *Medium*.

Mitchell, A., Shearer, E., & Stocking, G. (2021). News on Twitter: Consumed by Most Users and Trusted by Many. *Pew Research*.

RITHP. (2023). Logistic Regression and regularization: Avoiding overfitting and improving generalization. *Medium*.

Roos, D. (2022). The Deadliest Events in US History. *History Channel*.

Setia, M. (2023). Binary Cross Entropy aka Log Loss-The cost function used in Logistic Regression. *Analytics Vidhya*.