Malicious Tweet Identifier

Alek Racz, Steven Hurley, Jonathon Wright, Matt Robinson, Nicholas Jones, Marcus Summers

Sponsor: Kaushik Madala

Purpose

With the exponential rise of misinformation and data consumed by average users, it is important to make sure that Internet users are educated on how to identify malicious or garbage data.

This app is designed to be an educational resource to help identify tweets, Twitter profiles, or hashtags as being questionable. There are also resources on the website to help train the user to better identify malicious tweets.

We made it accessible via a web app to ensure that it is easy for anyone to access and learn about the dangers and prevalence of garbage information on the Internet.

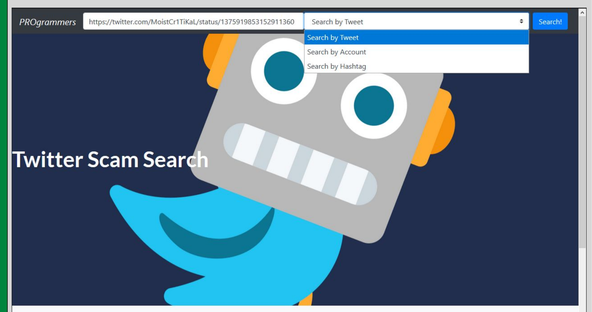
Usage

The user is able to input a tweet URL and receive an identifier and a confidence rating based according to our AI. In addition to individual tweets, profiles and hashtags can be scanned and will also be given an average rating for example, a hashtag can be identified as having 20% malicious tweets of 500 scanned tweets.

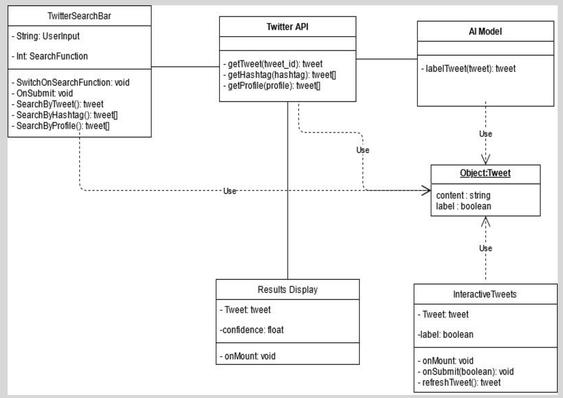
Technologies

mongoDB. · node · express · React · Flask · NGINX · DigitalOcean · docker

Screenshots



Class Diagram



Architecture

The project is built using 3 containers each responsible for its own environment. We have a frontend container who hosts our React driven site, a backend hosting the Express server and MongoDB, and a Flask container holding our AI model and Twitter API.

These three containers communicate between each other using RESTful APIs inside a Debian VM in a Digital Ocean Droplet.

AI Methodology

With our project we have trained two different models to differentiate between tweets we consider spam, and tweets which are not. We've defined spam tweets as tweets which promote a product or business not directly related to the account. This means an official Arby's account tweets are not spam, but the Arby's App tweeting on behalf of normal users are.

To train a model to recognize this we collected around five thousand tweets, half of which are what we define as spam tweets and the other half are not. This data is labeled and processed to be read in by the models.

To normalize the data we remove links, we lowercase the text, we remove any emojis which may be present, as well as tag the part of speech for every word and lemmatize words where appropriate. We also remove stopwords, which don't provide us any useful information.

After the tweets have all finished preprocessing we look at Term frequency and remove very frequent words, much as we do stop words using a vectorizer. Once these words are removed the vectorizer essentially creates a vector out of each tweet from the wordcount of the tweet. This is effective because the language of spam tweets uses a specific vocabulary is far more limited than that of normal tweets.

AI Models

In this project we train two separate models to identify spam tweets. With two models we can create a confidence score of how sure we are that a given tweet is spam or not, as if both models make the same prediction we can say with a higher degree of certainty than if not.

The models used are Naive Bayes Classifier, and Support Vector Machine. These models each have their own strengths so considering the output for both of them on every prediction is valuable and only serves to enrich our predictions.

Retrospective

Overall, this project was a lot of fun building out. Being a full-stack project, everyone on the team got to contribute into fields they were interested in.

Given the opportunity to restart from the beginning, we definitely would have put software testing at the forefront of our development. We also believe having stronger roles and feature assignments in the team would have led to faster development of core features.

Agile methodologies allowed us to refine and redirect our development where we needed it to go between sprints and helped a lot in keeping our goals in sight.

React provided a quick and easy way to rapid prototype and build our website. The turnaround time to build our react components was shockingly short and very straightforward code-wise.

The most surprising obstacle in our project was learning that it violates twitters TOS to share the contents of tweets outside of twitter. This meant finding data was quite difficult, and that tweets could only be shared via the tweet ID and not the actual text.

Successes

We found that both the Naives Bayes Classifier and the Support Vector Machine performed at a rate which validated our hypotheses. The language of spam tweets is similar enough to be identified via machine learning.

Model	Accuracy
Naive Bayes Classifier	**82.4%**
Support Vector Machine	**84.7%**

We were able to host all of our environments inside containers using a cloud service and successfully fulfill end user requests without any interruptions.

React provided a great framework for quickly developing functional pieces of what would have been a static website. Our front end team were able to make several components that made the website feel more cohesive.