Machine Learning Engineer Nanodegree Capstone Proposal

Real or Not? NLP with Disaster Tweets Challenge

Akihiro Nomura
January 3rd, 2020

Proposal

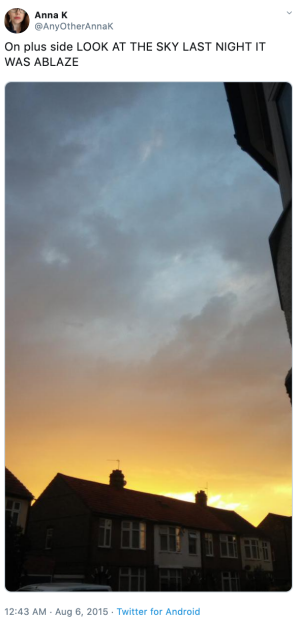
Domain Background

Twitter has become an important communication channel in times of emergency. The ubiquitousness of smartphones enables people to announce an emergency they're observing in real-time. Because of this, more agencies are interested in programatically monitoring Twitter (i.e. disaster relief organizations and news agencies).

This project was inspired by Kaggle competition (https://www.kaggle.com/c/nlp-getting-started/overview).

Problem Statement

Unfortunately, It's not always clear whether a person's words are actually announcing a disaster. Please see the right example. The author explicitly uses the word "ABLAZE" but means it metaphorically. This is clear to a human right away, especially with the visual aid. But it's less clear to a machine.



The goal is to build a machine learning model that predicts which tweets are about real disasters and which one's aren't. So, I'll try to solve a kind of natural language processing (NLP) problem.

Tweet source: https://twitter.com/AnyOtherAnnaK/status/629195955506708480

Datasets and Inputs

For this project, I can use a dataset of 10,000 tweets that were hand classified. Each entry has the following information. You can download the data from this link (https://www.kaggle.com/c/nlp-getting-started/data).

- id - a unique identifier for each tweet
- text - the text of the tweet
- location - the location the tweet was sent from (may be blank)
- keyword - a particular keyword from the tweet (may be blank)
- target - this denotes whether a tweet is about a real disaster (1) or not (0)

Solution Statement

This is a supervised learning problems, because the dataset is labeled. I'd like to build a binary classfication model which is used to predict whether a given tweet is about a real disaster or not.

Benchmark Model

As a benchmark model, I plan to use a XGBoost model. XGBoost stands for extreme gradient boosting, which is an implementation of gradient boosting with several additional features focused on performance and speed. With careful parameter tuning, we can train highly accurate models.

Evaluation Metrics

A model in this project is evaluated using F1-score between the predicted and expected answers. F1-score, a measure of a test's accuracy, can be interpreted as a weighted average of the precision and recall. F1-score is calculated as follows:

$$F1 = 2 * \frac{precision * recall}{precision + recall} = \frac{2TP}{2TP + FP + FN}$$

where:

$$precision = \frac{TP}{TP + FP}, \quad recall = \frac{TP}{TP + FN}$$

TP, FN and FP represent the number of true positives, false negatives and false positives, respectively.

Project Design

Programming language: Python 3.6
Library: Pandas, Matplotlibs, Numpy, NLTK (Natural Language Toolkit), Scikit-learn, XGBoost
Workflow:

1. Loading and exploring the data
2. Pre Processing and cleaning the data
    - Removing emojis
    - Removing punctuations
    - Spelling Correction, and so on...
3. Building and training a model
4. Making improvements on the model
5. Evaluating and comparing model test performance