**Capstone Proposal**

Duong Nguyen  
July 20th, 2021

**Domain Background**

Twitter, which is one of the most impactful social networks, has provided the facilities for millions of users to share information and news. Tweets reflect personal opinions about different topics. The ability to extract and collect information from the tweets is of great benefits for commercial as well as personal purposes. For example, collecting the tweets related to a product can help the company for the marketing campaign. However, there is a massive amount of data posted daily in the form of short texts or audio. It is nearly impossible to do the analytics manually. Hence, there has been many research trying to do the semantic analysis with tweets automatically. One of the recent works on the tweet topic is <https://www.hindawi.com/journals/complexity/2020/8892552/>. In the paper, the authors tried to classify and do sentiment analysis on tweets related to healthcare. The aim of the research is to recommend users with personalized tweets. Techniques in Natural language processing (NLP) are used in the paper. NLP is a branch of Artificial Intelligent, which exploits rules, statistic, and neural network techniques to process and analyse large amount of natural language data. NLP has been used to create many useful applications such as spam filter, online translators, text analytics, etc.

**Problem Statement**

Inspired by that, this project also works on tweet classification but in another field. In particular, the project is going to predict if a tweet mentions about a real disaster or not. The dataset contains of the tweets about disasters and random tweets. The result can be used to detect a surging number of tweets related to disaster and provide on time alerts to government or people in the affected areas. That could help to trigger preventive methods faster and reduce the destructive level. NLP techniques are used here.

**Datasets and Inputs**

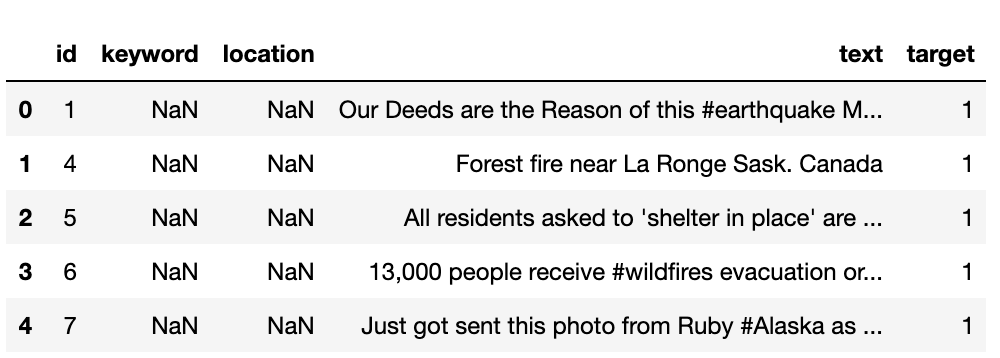
The datasets are from a competition in Kaggle. <https://www.kaggle.com/c/nlp-getting-started>

There are two datasets. One is the training set, and the other is the test set. The datasets are given as csv files. The data inside the csv files can be loaded into the dataframe. The training set contains of 7613 samples. The test set contains of 3263 samples. The training set is used to build the models. The test set will be used for prediction and submitted the result to Kaggle for evaluation

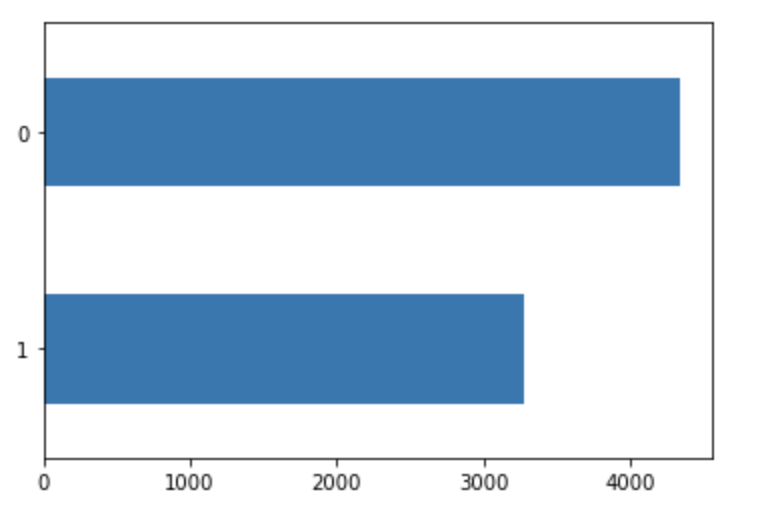
The attributes for each sample are:

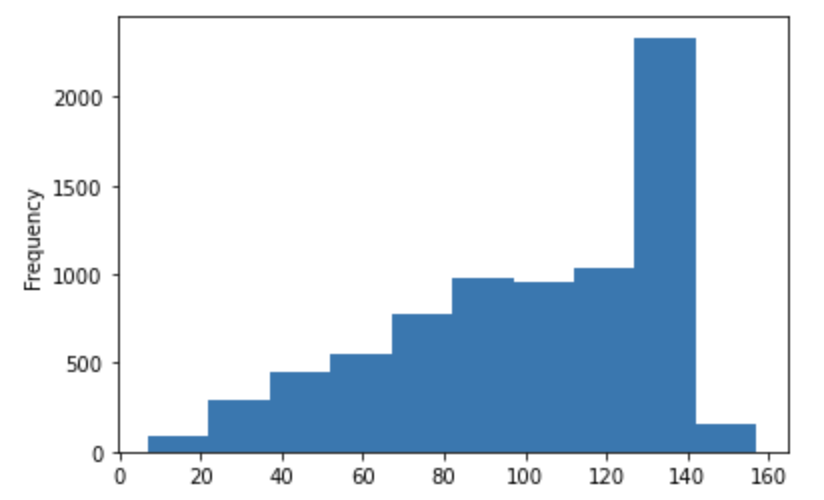
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id** | **Keyword** | **Location** | **Text** | **Target** |
| Id of the row | keyword that determines if the text mentions about the disaster | Location of the tweets | The message of the tweets | 1 if this is a tweet about disaster, 0 if this is not. This attribute is only available in training set. |

**Snapshot of the data:**

****

The train dataset is quite balanced. The number of positive (target=1) and negative (target=0) samples do not differ significantly.



Also, the length of the text ranges from 10 to 160 characters.

**Solution Statement**

This is the classification problem. The feature (X) is the text column, and the target variable (Y) is the target column. The feature of each sample (row) is cleaned and convert to an array of number using TfidfTransformer. CountVectorizer is not a good solution to extract the features because the most common words are I, and, the, etc. which cannot be used to predict the tweets.

Two models are developed to solve this problem. One is the neutral network built with Pytorch, and the other is XGBoosting. Those are the popular models for classification problems.

The results from both models are submitted to Kaggle to get the better model.

**Benchmark Model**

A logistic regression model is picked as the benchmark model: <https://www.kaggle.com/hemanthkumarkarangi/logistic-model-disaster-tweet>.

That is because logistic regression is basic algorithm to train a model. The benchmark solution is able to use a simple algorithm to achieve a public score of 0.79, which is higher than some models built with complex algorithm such as LSTM or Neutral network. In this project, XGBoost and Neural network are used to train the models with the hope that a better result can be achieved with more complex algorithms.

**Evaluation Metrics**

The evaluation metric is F1. The result will be submitted to Kaggle and compared with other submission.

**Project Design**

The project contains of five parts

1. Data analytics

In this part, the dataset will be analyzed. The target values are plotted to see the distribution of the values. There are classes for the target: 1 and 0, and the data is almost balanced.

1. Data preprocessing

In this part, data is cleaned. Various techniques are applied on the dataset to standardize the input. The techniques used include removing HTML elements, lowering text, stemmer, removing shortened strings, etc.

1. Feature extractions

CountVectorizer is used to analyze the word in the texts. The most frequency words are I, the, you, etc. and they are not the keywords. Hence, TfidfTransformer is used to extract features from the text.

1. Model development

Data is trained with XGBoosting and Pytorch models. XGBoosting gives higher results, so is used as the final model.

1. Model evaluation

The predicted values are submitted to Kaggle for evaluation. The final score is: 0.77