**Tittle**

Disaster Tweet Classification via Natural language processing

**Problem**

To identify whether a person’s words are actually announcing a disaster or not

**Data**

Examples of some of the data from twitter

-I can see a fire in the woods...

-There's an emergency evacuation happening now

I can see a fire in the woods...

**Methods/ Approach**

I followed 2 different approaches in this project.

1. *Bag-of-words Approach:*

One hot encode text data & then apply traditional machine learning models (used Logistic Regression & XGBoost)

1. *Word Embedding Approach:*
   1. Generate word embedding from the training dataset
   2. Using pretrained word embeddings

Apply Logistic Regression & Multi-Layer Perceptron on word vectors to classify

**Results**

Logistic Regression f1\_score: 0.75

Logistic Regression Accuracy: 0.81

Logistic Regression Precision: 0.84

Logistic Regression Recall: 0.69

Using Logistic Regression, I was able to identify 69% of disaster tweets with 81% accuracy

**Analysis, graphs**

**Text

Description automatically generated**

**Chart, line chart

Description automatically generated**

**Conclusion:**

This is a good starting point using vanilla methods. Implementing other text cleaning techniques like spelling correction & employing RNN may improve accuracy of the process.

**Source Code**

**Git hub link: Source Code**

https://github.com/var2019/Data-science-project--Varsha

**References:**

<https://mlexplained.com/2018/02/08/a-comprehensive-tutorial-to-torchtext/>

<https://gist.github.com/slowkow/7a7f61f495e3dbb7e3d767f97bd7304b>

<https://www.dataquest.io/blog/tutorial-text-classification-in-python-using-spacy/>

<https://edumunozsala.github.io/BlogEms/jupyter/nlp/classification/embeddings/python/2020/08/15/Intro_NLP_WordEmbeddings_Classification.html>