Natural Language Processing with Disaster Tweets

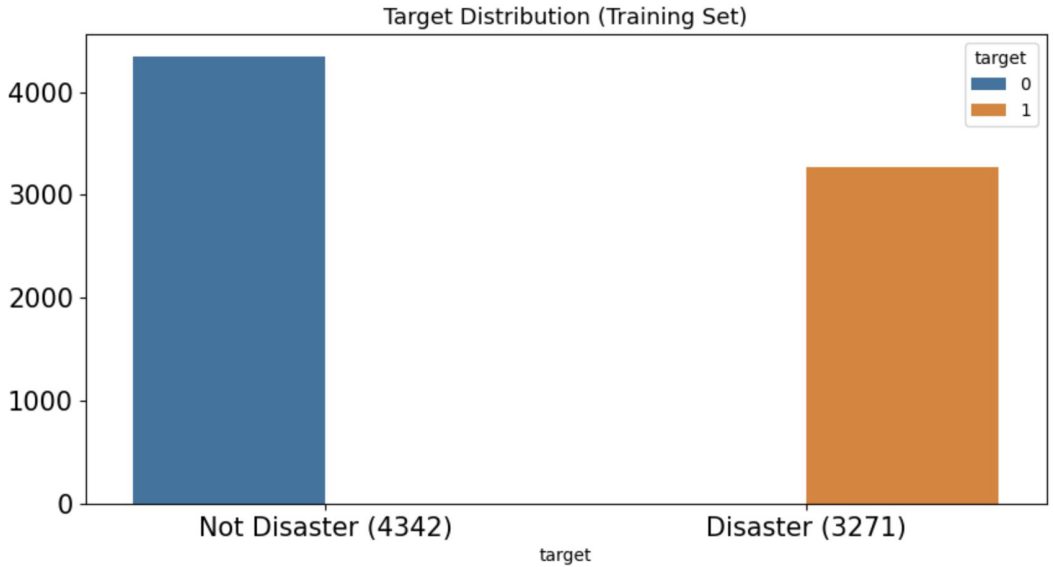
Anubha Nagar, Zhuoyi Zhan, Dara Kuno

Project Goal

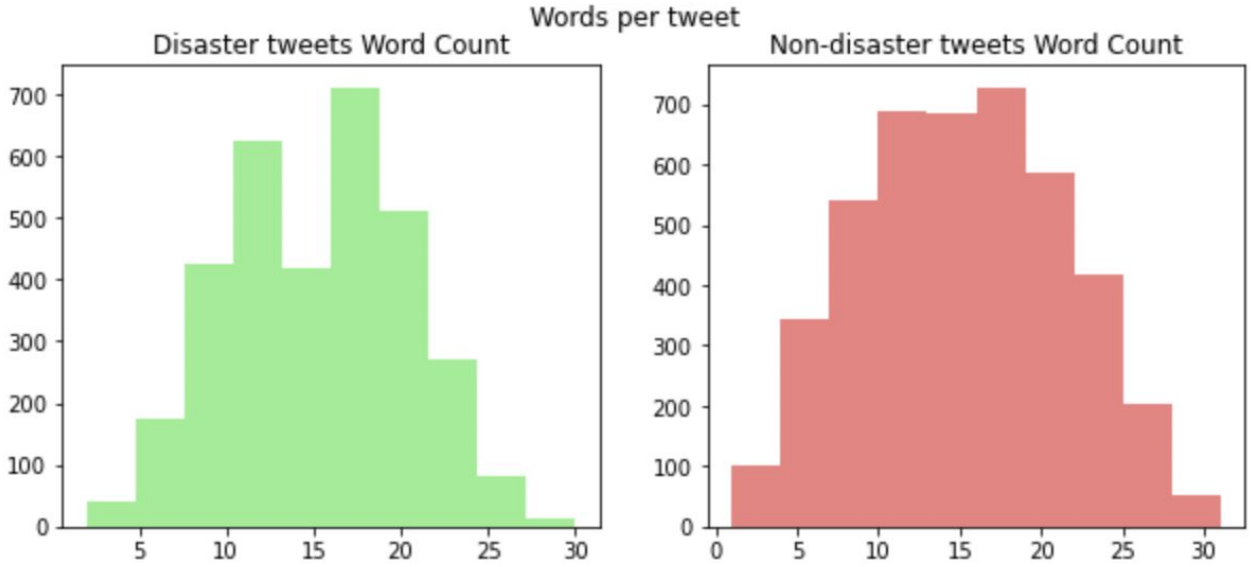
The goal is to build a machine learning model that predicts which Tweets are about real disasters and which ones aren’t

Exploratory Data Analysis

Target Distribution



Word Count of Dataset



Mean word count:

1- 15.167532864567411

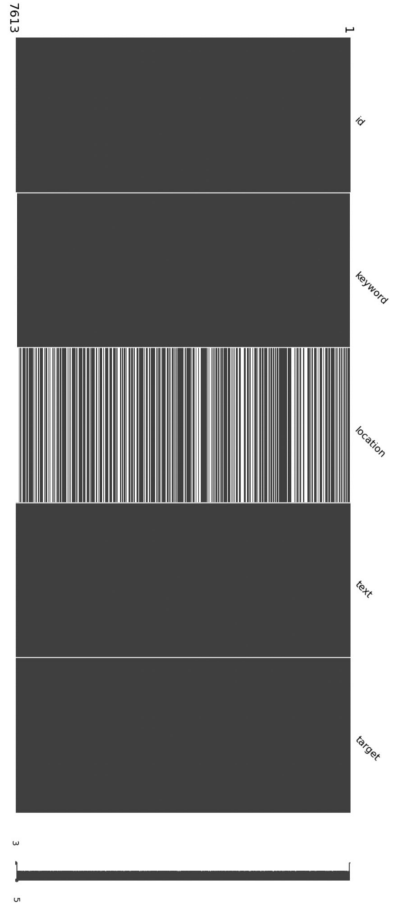
0 - 14.704744357438969

NA values

Columns	Missing Data Count
id	0
keyword	61
location	**2533**
text	0
target	0

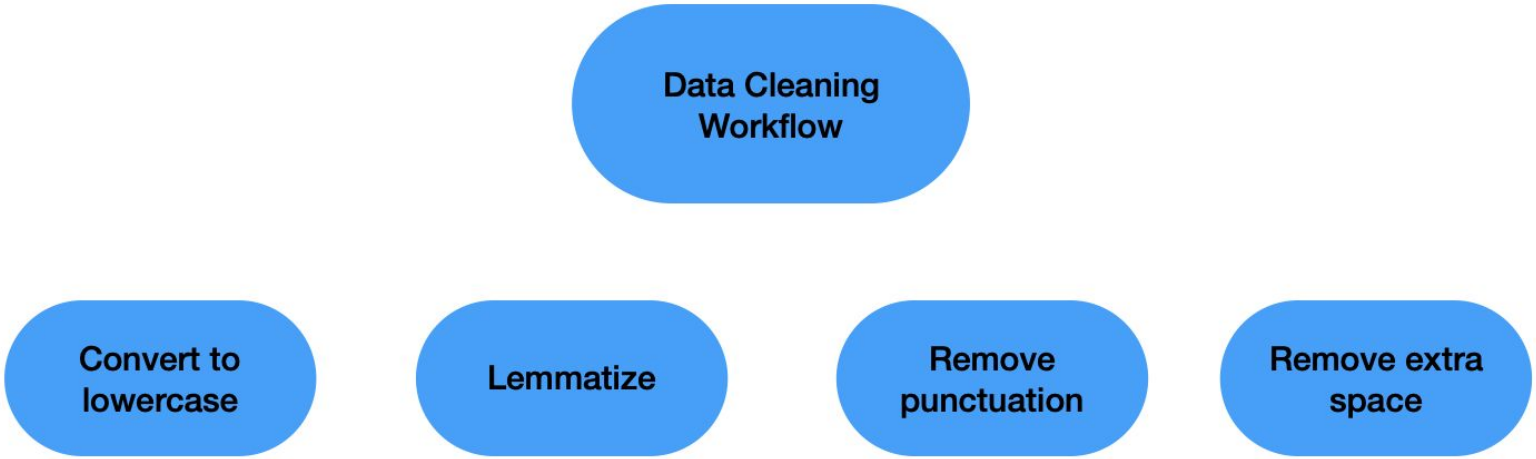
We use this

We use this



Data Cleaning

Data Cleaning Workflow



Data Preparation before modeling

Data Preparation

- Used auto-tokenizers specific to the model eg: BertTweet-base
    - NLTK toolkit
    - In this package, used the emoji package to translate emojis to text strings
    - Convert url and mentions to special tokens (@USER and HTTPURL)

Modeling

Data Splitting

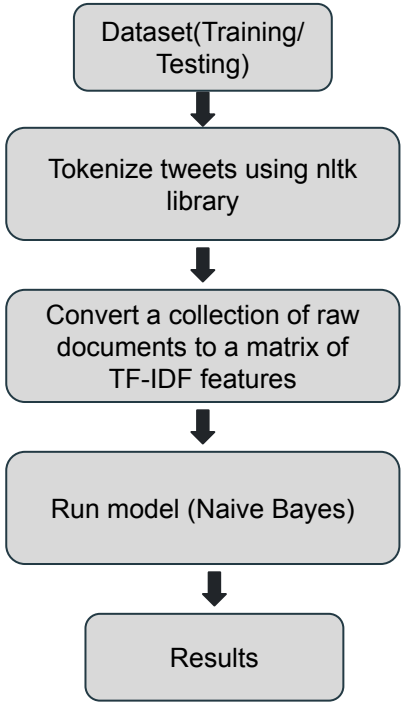
Data	Percent
Training	85%
Validation	7.5%
Testing	7.5%

Baseline Model: Naive Bayes

Why Naive Bayes?

- Naive Bayes is a good algorithm for working with text classification since it performs better than other models with less training data is the assumption of independence of features holds
- Naive Bayes can adapt quickly to the changes and new data.
- Naive Bayes model is a fast and simple classification algorithms that are suitable for very high-dimensional datasets. It have so few tunable parameters, so is useful as a quick-and-dirty baseline for classification.

Naive Bayes Method



	precision	recall	f1-score
0	0.76	0.93	0.83
1	0.88	0.62	0.73
accuracy			0.79

Bert-Base Uncased Transformer Model

Baseline

- Learning Rate = 0.0001
- Activation Function: Sigmoid
- Batch Size = 32
- Optimizer: Stochastic Gradient Descent
- Accuracy: **76%**

Parameter tuning of the model

Activation Function	Optimizer	Batch size	Learning rate	Accuracy	Training Loss	Validation Loss	Validation Accuracy	Testing Accuracy
Sigmoid	Adam	32	0.001	**0.81**	**0.43**	**0.42**	**0.81**	**0.78**
Sigmoid	SGD	32	0.001	0.79	0.46	0.49	0.77	0.76
Tanh	Adam	32	0.001	0.65	0.65	0.68	0.69	0.70
Relu	Adam	32	0.001	0.69	0.89	0.80	0.73	0.56
Leaky ReLU	Adam	32	0.001	0.73	0.69	0.65	0.73	0.73

BertTweet Model

BERTweet Baseline

- Cleaning and preprocessing:
  - Lowercase and basic contractions
  - Basic emojis
- Pretrained Vinai Bertweet-Base model
- epoch : 2

	Accuracy	Training Loss	Validation Accuracy	Validation Loss	Testing Accuracy	Testing rocauc	Testing f-1
1	0.82	0.28	0.84	0.42	0.81	0.81	0.78

BertTweet Model Parameter-tuning

BERTweet Finetune

- Pretrained Vinai Bertweet-Base model
- Activation = relu, sigmoid
- Optimizer: Adam
- Loss: binary cross entropy
- Metrics: accuracy and f1 score
- Epoch : 2

Parameters used

```
grid = [{'hidden_n': 16, 'drop': 0.3, 'lr': 1e-5, 'weight_decay': 1e-6},
        {'hidden_n': 32, 'drop': 0.3, 'lr': 1e-5, 'weight_decay': 1e-6},
        {'hidden_n': 32, 'drop': 0.3, 'lr': 5e-6, 'weight_decay': 1e-6},
        {'hidden_n': 32, 'drop': 0.25, 'lr': 1e-5, 'weight_decay': 5e-6}
        {'hidden_n': 32, 'drop': 0.35, 'lr': 1e-5, 'weight_decay': 5e-7}
        {'hidden_n': 64, 'drop': 0.3, 'lr': 1e-5, 'weight_decay': 1e-6},
        {'hidden_n': 64, 'drop': 0.3, 'lr': 1e-5, 'weight_decay': 5e-6},]
```

Results

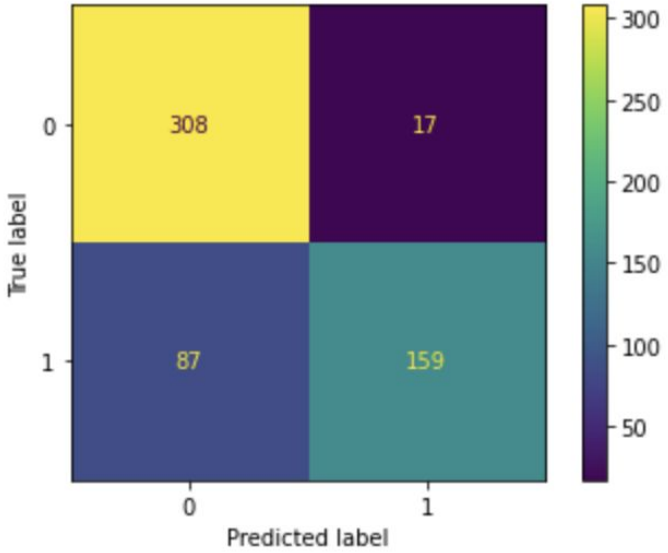
	Accuracy	Training Loss	f-1	Validation Loss	Validation Accuracy	validation f-1
1	0.86	0.37	0.82	0.39	0.85	0.82
2	0.79	0.50	0.69	0.46	0.84	0.79
3	0.84	0.54	0.80	0.51	0.85	0.81
4	**0.87**	**0.47**	**0.83**	**0.43**	**0.85**	**0.82**
5	0.86	0.50	0.82	0.46	0.85	0.82
6	0.87	0.45	0.84	0.43	0.85	0.82
7	0.88	0.42	0.85	0.41	0.85	0.81

Testing Results

Testing accuracy: 0.82

F1_score 0.75

Roc_auc_score 0.797



Future Work

- We can create an API design where we enter the tweet and it shows if it is a natural disaster or not
- Think of ways we can incorporate keyword
- Try other BertTweet models

Thank you!

Github repo link:

https://github.com/darasliwinski/nlp_disaster