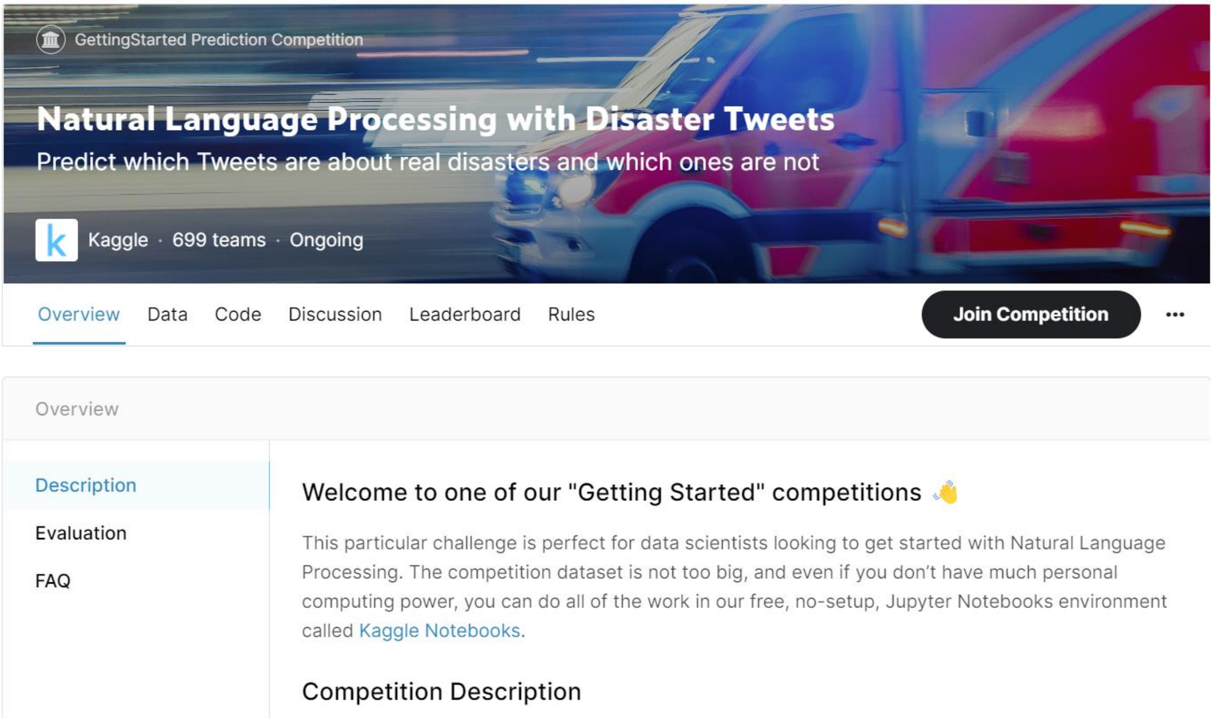
Natural Language Processing with Disaster Tweets

Team Member List:
Aaditya Damle (UTA ID: 1001955625)
Rutuja Dukhande(UTA ID: 1001730748)
Mohit Somaiya(UTA ID: 1001950441)
Avijit Tripathi(UTA ID: 1001937928)

Introduction:

Natural Language Processing with Disaster Tweets:
We worked on a Kaggle competition with Natural Language Processing.
Twitter has grown to be a crucial communication tool during emergencies. Smartphones are so common that anyone can instantly report an emergency they are witnessing. As a result, more organizations are interested in automating Twitter monitoring (i.e. disaster relief organizations and news agencies).
However, it's not always obvious if someone is genuinely foreseeing a catastrophe when they speak.
We are required to create a machine learning model in this competition that can determine which Tweets are about actual disasters and which ones aren't.
A dataset of 10,000 tweets that were manually categorized will be made available.

Here is the competition link: https://www.kaggle.com/competitions/nlp-getting-started



GettingStarted Prediction Competition

Natural Language Processing with Disaster Tweets

Predict which Tweets are about real disasters and which ones are not

Kaggle · 699 teams · Ongoing

Welcome to one of our "Getting Started" competitions 👋

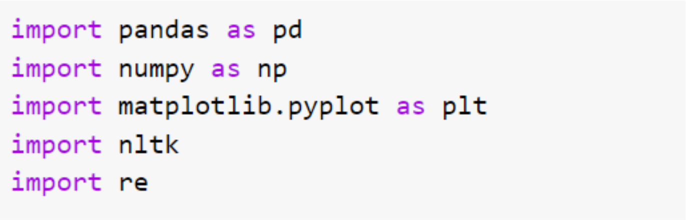
This particular challenge is perfect for data scientists looking to get started with Natural Language Processing. The competition dataset is not too big, and even if you don't have much personal computing power, you can do all of the work in our free, no-setup, Jupyter Notebooks environment called Kaggle Notebooks.

Competition Description

NLP:

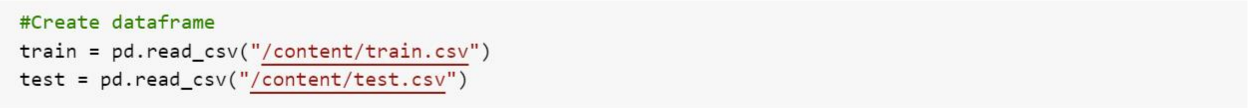
The practice of using software or a machine to manipulate or comprehend speech or text is known as natural language processing (NLP). Human interaction, understanding of one another's viewpoints, and providing the proper response are examples of an analogy. In NLP, a computer performs this interaction, comprehension, and response in place of a human.

Libraries imported to work on NLP

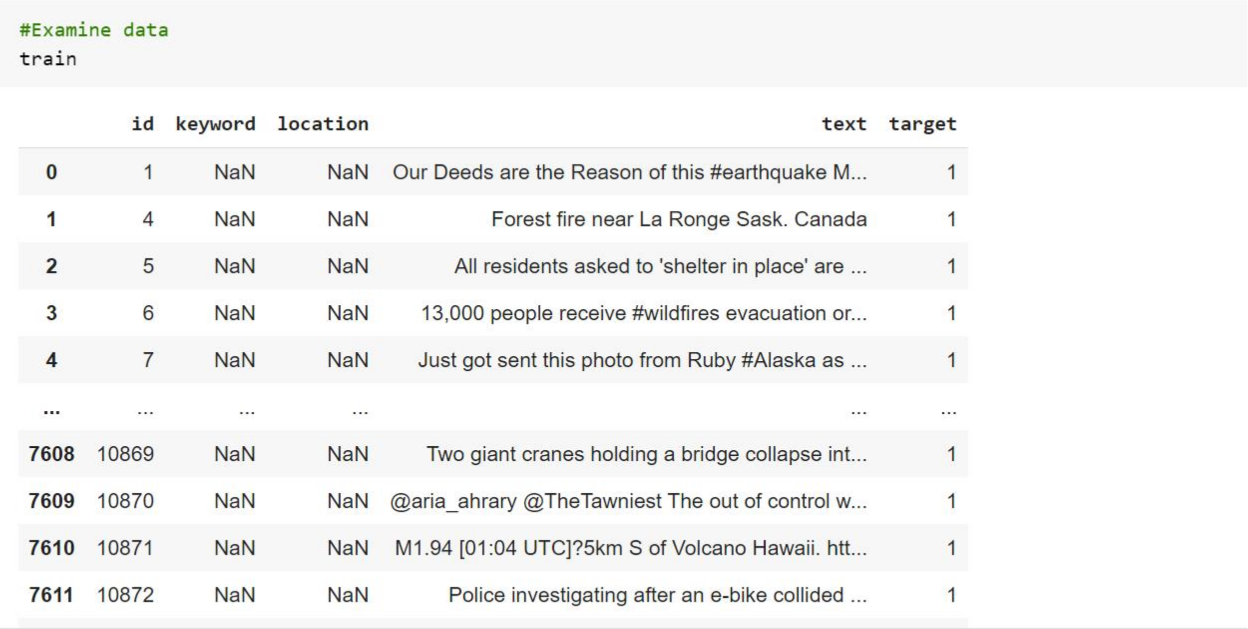


```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import nltk
import re
```

Dataset



```python
#Create dataframe
train = pd.read_csv("/content/train.csv")
test = pd.read_csv("/content/test.csv")
```

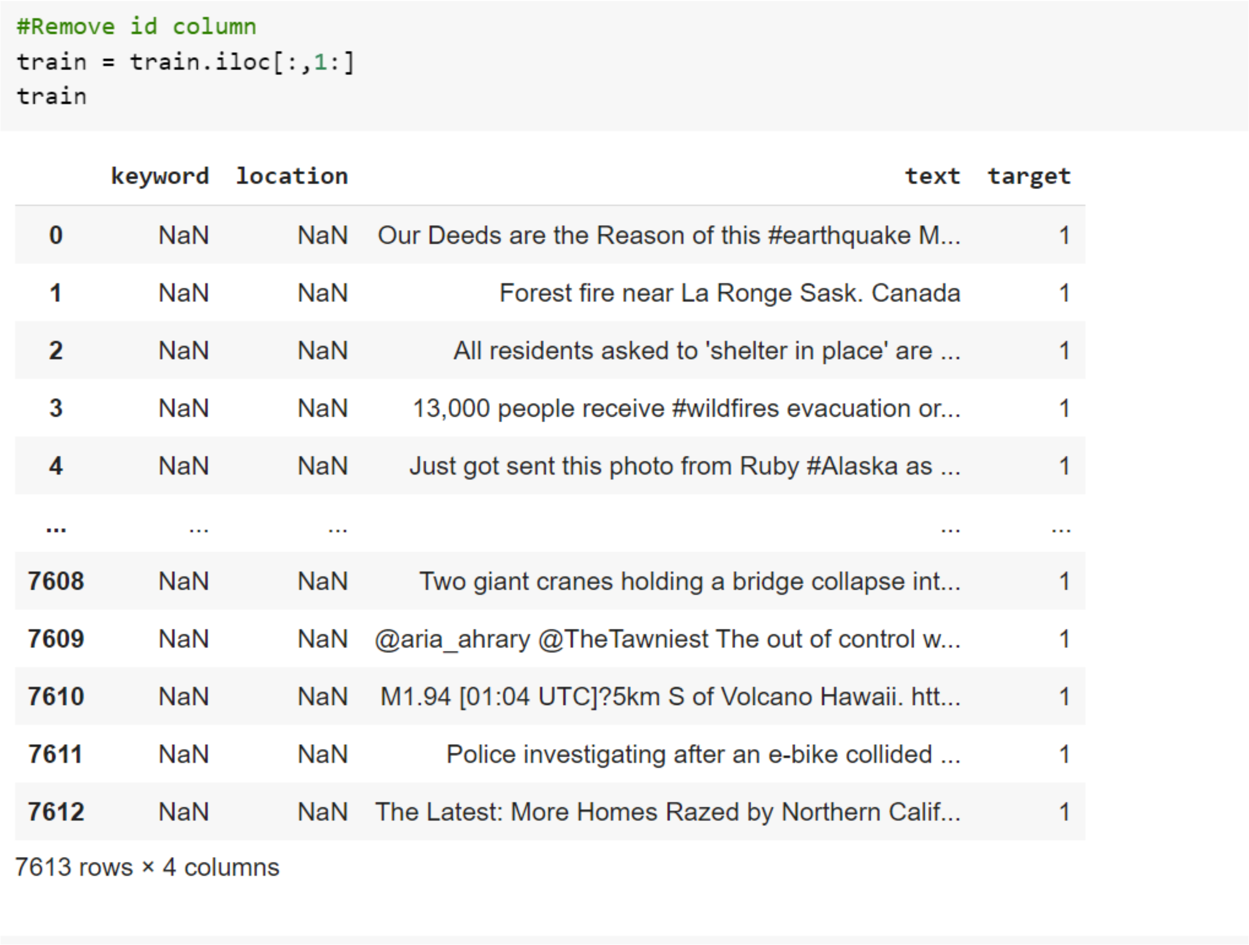


```python
#Examine data
train
```

	id	keyword	location	text	target
0	1	NaN	NaN	Our Deeds are the Reason of this #earthquake M...	1
1	4	NaN	NaN	Forest fire near La Ronge Sask. Canada	1
2	5	NaN	NaN	All residents asked to 'shelter in place' are ...	1
3	6	NaN	NaN	13,000 people receive #wildfires evacuation or...	1
4	7	NaN	NaN	Just got sent this photo from Ruby #Alaska as ...	1
...	...	...	...	...	...
7608	10869	NaN	NaN	Two giant cranes holding a bridge collapse int...	1
7609	10870	NaN	NaN	@aria_ahrary @TheTawniest The out of control w...	1
7610	10871	NaN	NaN	M1.94 [01:04 UTC]?5km S of Volcano Hawaii. htt...	1
7611	10872	NaN	NaN	Police investigating after an e-bike collided ...	1

Data Preprocessing:

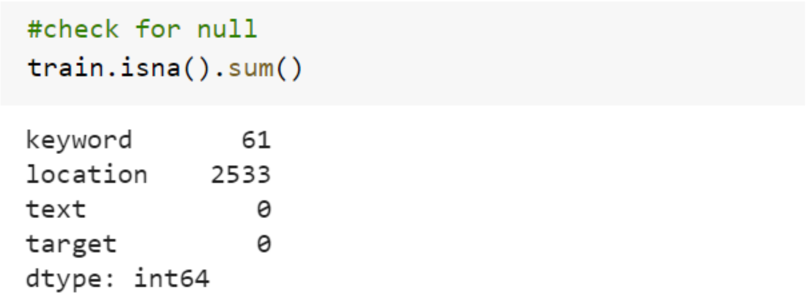
We did some data cleaning as it is very necessary to perform data cleaning for NLP tasks.We removed id columns, removed null values, etc.



```
#Remove id column
train = train.iloc[:,1:]
train
```

	keyword	location	text	target
0	NaN	NaN	Our Deeds are the Reason of this #earthquake M...	1
1	NaN	NaN	Forest fire near La Ronge Sask. Canada	1
2	NaN	NaN	All residents asked to 'shelter in place' are ...	1
3	NaN	NaN	13,000 people receive #wildfires evacuation or...	1
4	NaN	NaN	Just got sent this photo from Ruby #Alaska as ...	1
...	...	...	...	...
7608	NaN	NaN	Two giant cranes holding a bridge collapse int...	1
7609	NaN	NaN	@aria_ahrary @TheTawniest The out of control w...	1
7610	NaN	NaN	M1.94 [01:04 UTC]?5km S of Volcano Hawaii. htt...	1
7611	NaN	NaN	Police investigating after an e-bike collided ...	1
7612	NaN	NaN	The Latest: More Homes Razed by Northern Calif...	1

7613 rows × 4 columns



```
#check for null
train.isna().sum()
```

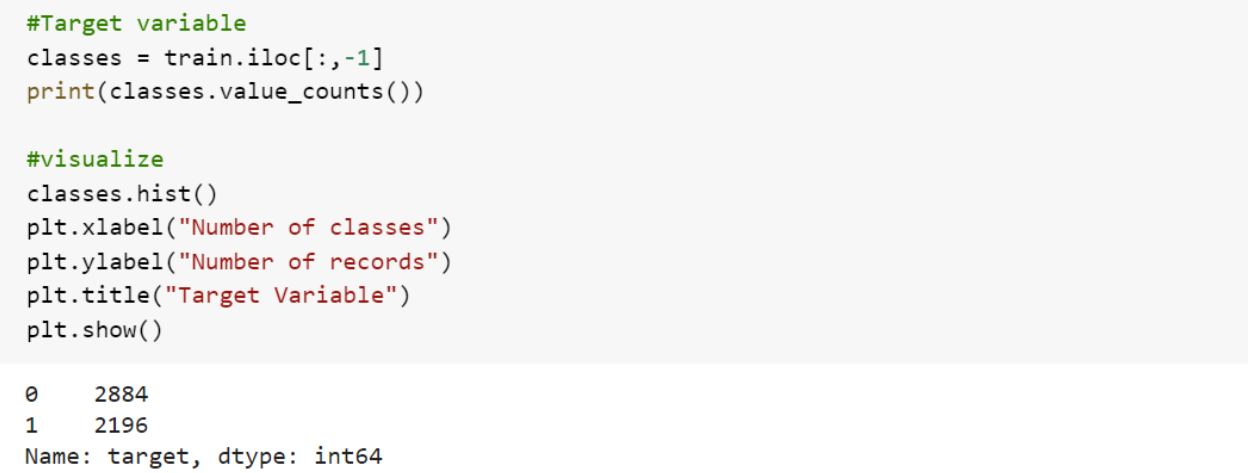
```
keyword       61
location    2533
text           0
target         0
dtype: int64
```

```
#remove na values
train = train.dropna()
train
```

	keyword	location	text	target
31	ablaze	Birmingham	@bbcmtd Wholesale Markets ablaze http://t.co/l...	1
32	ablaze	Est. September 2012 - Bristol	We always try to bring the heavy. #metal #RT h...	0
33	ablaze	AFRICA	#AFRICANBAZE: Breaking news:Nigeria flag set a...	1
34	ablaze	Philadelphia, PA	Crying out for more! Set me ablaze	0
35	ablaze	London, UK	On plus side LOOK AT THE SKY LAST NIGHT IT WAS...	0
...	...	...	...	...
7575	wrecked	TN	On the bright side I wrecked http://t.co/uEa0t...	0
7577	wrecked	#NewcastleuponTyne #UK	@widda16 ... He's gone. You can relax. I thoug...	0
7579	wrecked	Vancouver, Canada	Three days off from work and they've pretty mu...	0
7580	wrecked	London	#FX #forex #trading Cramer: Iger's 3 words tha...	0
7581	wrecked	Lincoln	@engineshed Great atmosphere at the British Li...	0

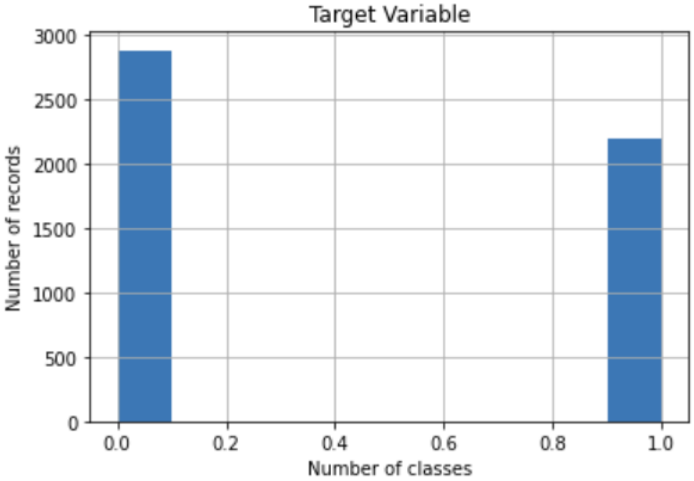
5080 rows × 4 columns

Visualization of Target Variable:

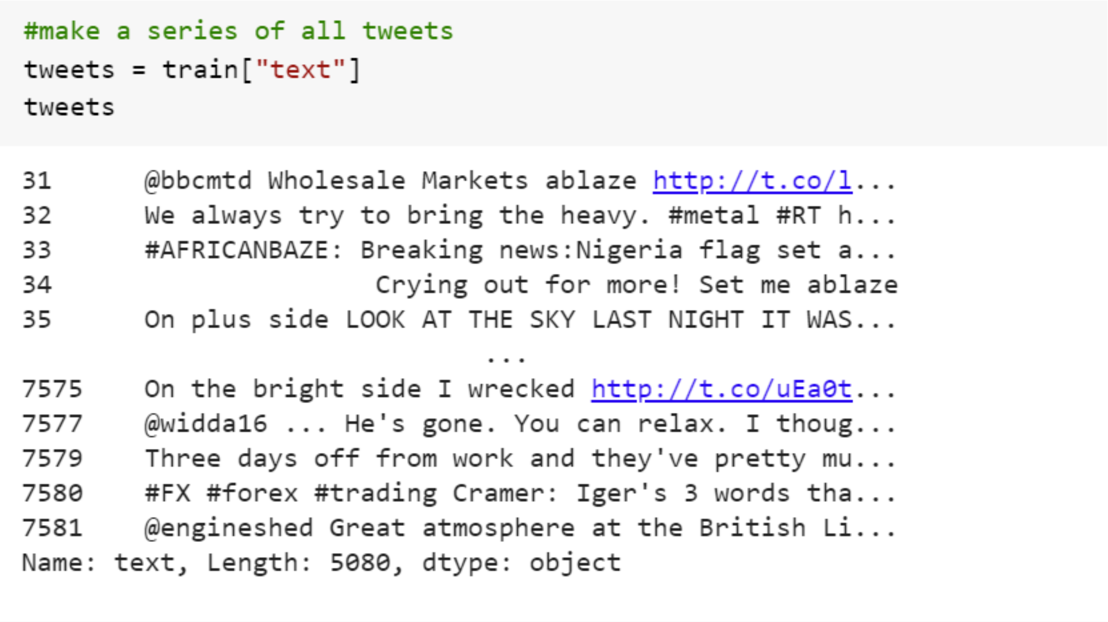


```
#Target variable
classes = train.iloc[:,-1]
print(classes.value_counts())

#visualize
classes.hist()
plt.xlabel("Number of classes")
plt.ylabel("Number of records")
plt.title("Target Variable")
plt.show()
```

```
0    2884
1    2196
Name: target, dtype: int64
```

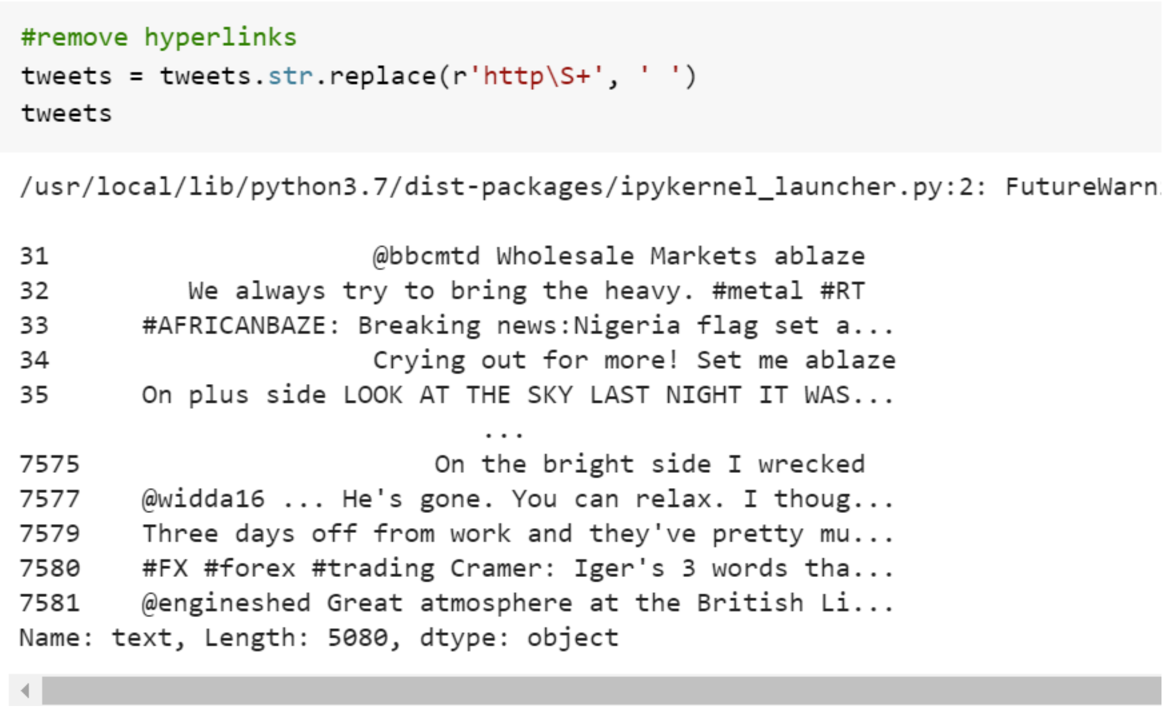


Pre-processing of Text



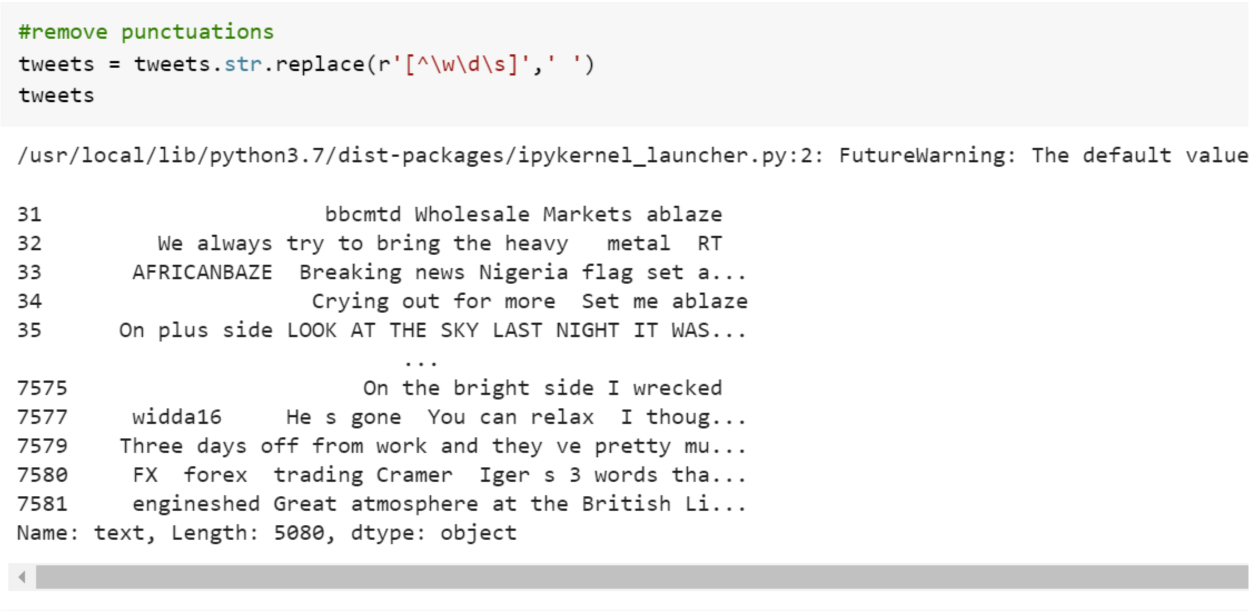
```
#make a series of all tweets
tweets = train["text"]
tweets
```

```
31      @bbcmtd Wholesale Markets ablaze http://t.co/l...
32      We always try to bring the heavy. #metal #RT h...
33      #AFRICANBAZE: Breaking news:Nigeria flag set a...
34                     Crying out for more! Set me ablaze
35      On plus side LOOK AT THE SKY LAST NIGHT IT WAS...
                                  ...
7575    On the bright side I wrecked http://t.co/uEa0t...
7577    @widda16 ... He's gone. You can relax. I thoug...
7579    Three days off from work and they've pretty mu...
7580    #FX #forex #trading Cramer: Iger's 3 words tha...
7581    @engineshed Great atmosphere at the British Li...
Name: text, Length: 5080, dtype: object
```



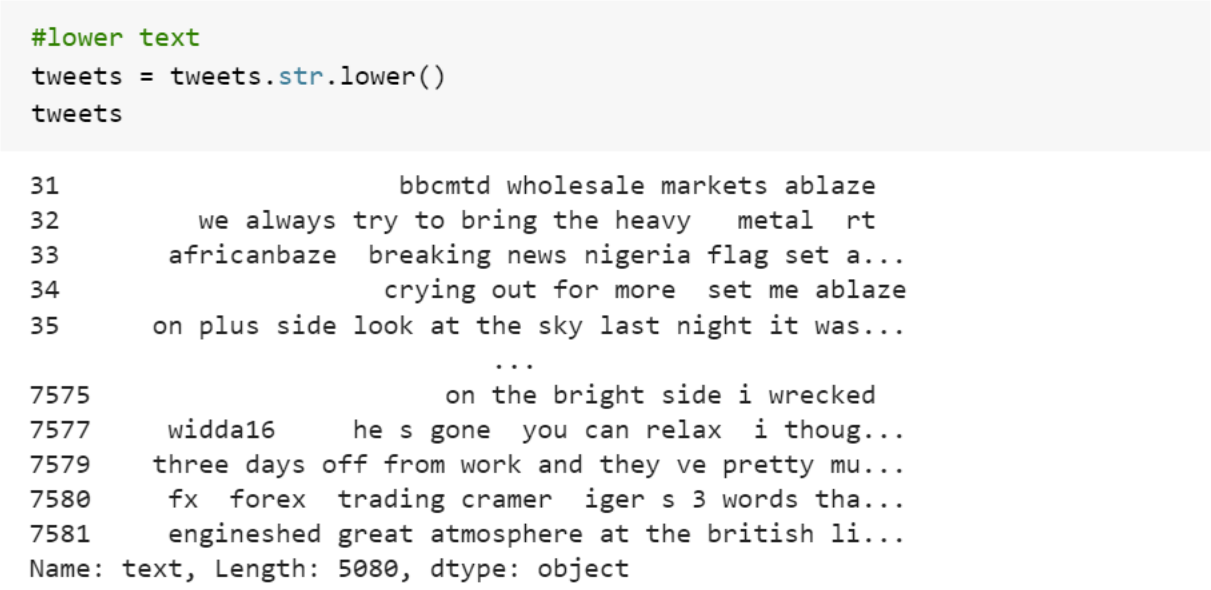
```
#remove hyperlinks
tweets = tweets.str.replace(r'http\S+', ' ')
tweets
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:2: FutureWarn

31                     @bbcmtd Wholesale Markets ablaze  
32         We always try to bring the heavy. #metal #RT  
33      #AFRICANBAZE: Breaking news:Nigeria flag set a...
34                     Crying out for more! Set me ablaze
35      On plus side LOOK AT THE SKY LAST NIGHT IT WAS...
                                  ...
7575                       On the bright side I wrecked  
7577    @widda16 ... He's gone. You can relax. I thoug...
7579    Three days off from work and they've pretty mu...
7580    #FX #forex #trading Cramer: Iger's 3 words tha...
7581    @engineshed Great atmosphere at the British Li...
Name: text, Length: 5080, dtype: object
```



```
#remove punctuations
tweets = tweets.str.replace(r'[^\w\d\s]',' ')
tweets
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:2: FutureWarning: The default value

31                      bbcmtd Wholesale Markets ablaze
32             We always try to bring the heavy   metal  RT
33           AFRICANBAZE  Breaking news Nigeria flag set a...
34                       Crying out for more  Set me ablaze
35          On plus side LOOK AT THE SKY LAST NIGHT IT WAS...
                                ...
7575                        On the bright side I wrecked
7577        widda16     He s gone  You can relax  I thoug...
7579       Three days off from work and they ve pretty mu...
7580        FX  forex  trading Cramer  Iger s 3 words tha...
7581        engineshed Great atmosphere at the British Li...
Name: text, Length: 5080, dtype: object
```



```
#lower text
tweets = tweets.str.lower()
tweets
```

```
31                      bbcmtd wholesale markets ablaze
32             we always try to bring the heavy   metal  rt
33           africanbaze  breaking news nigeria flag set a...
34                       crying out for more  set me ablaze
35          on plus side look at the sky last night it was...
                                ...
7575                        on the bright side i wrecked
7577        widda16     he s gone  you can relax  i thoug...
7579       three days off from work and they ve pretty mu...
7580        fx  forex  trading cramer  iger s 3 words tha...
7581        engineshed great atmosphere at the british li...
Name: text, Length: 5080, dtype: object
```

```
nltk.download("stopwords")
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
True
```

```
#stopwords
from nltk.corpus import stopwords
stop_words = set(stopwords.words("english"))
```

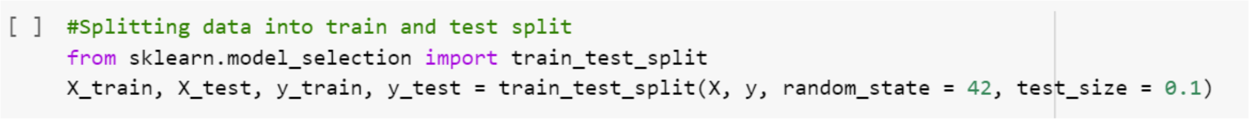
```
#removing common words
tweets = tweets.apply(lambda x: " ".join(word for word in x.split()
                                          if word not in stop_words))
tweets
```

```
31                        bbcmtd wholesale markets ablaze
32                        always try bring heavy metal rt
33      africanbaze breaking news nigeria flag set abl...
34                                      crying set ablaze
35                   plus side look sky last night ablaze
                              ...
7575                                  bright side wrecked
7577    widda16 gone relax thought wife wrecked cake g...
7579    three days work pretty much wrecked hahaha sho...
7580    fx forex trading cramer iger 3 words wrecked d...
7581    engineshed great atmosphere british lion gig t...
Name: text, Length: 5080, dtype: object
```

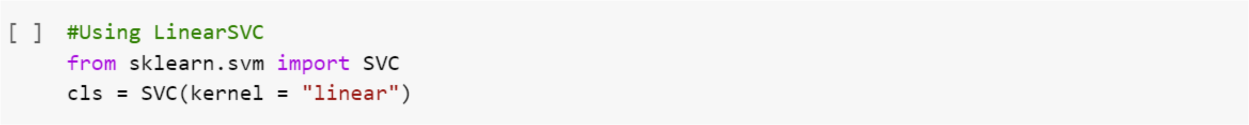
```
#Remove affixes to give stems using Porter Stemmer
ps = nltk.PorterStemmer()
tweets = tweets.apply(lambda x: ' '.join(ps.stem(word)
                      for word in x.split()))
tweets
```

```
31                            bbcmtd wholesal market ablaz
32                          alway tri bring heavi metal rt
33        africanbaz break news nigeria flag set ablaz aba
34                                           cri set ablaz
35                      plu side look sky last night ablaz
                                ...
7575                                     bright side wreck
7577    widda16 gone relax thought wife wreck cake gon...
7579    three day work pretti much wreck hahaha shouto...
7580    fx forex trade cramer iger 3 word wreck disney...
7581    enginesh great atmospher british lion gig toni...
Name: text, Length: 5080, dtype: object
```

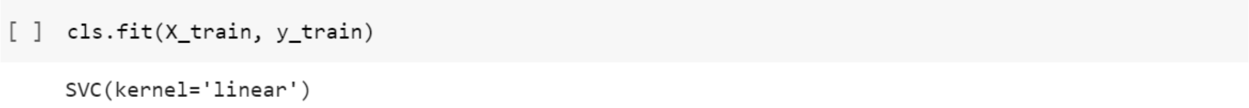
Model Training



```
[ ]  #Splitting data into train and test split
     from sklearn.model_selection import train_test_split
     X_train, X_test, y_train, y_test = train_test_split(X, y, random_state = 42, test_size = 0.1)
```

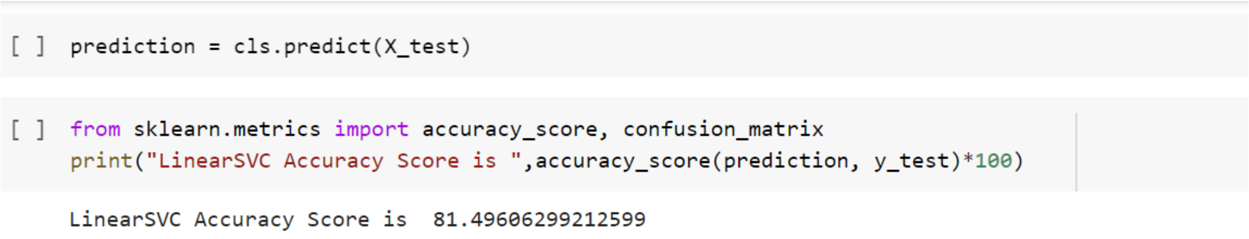


```
[ ]  #Using LinearSVC
     from sklearn.svm import SVC
     cls = SVC(kernel = "linear")
```



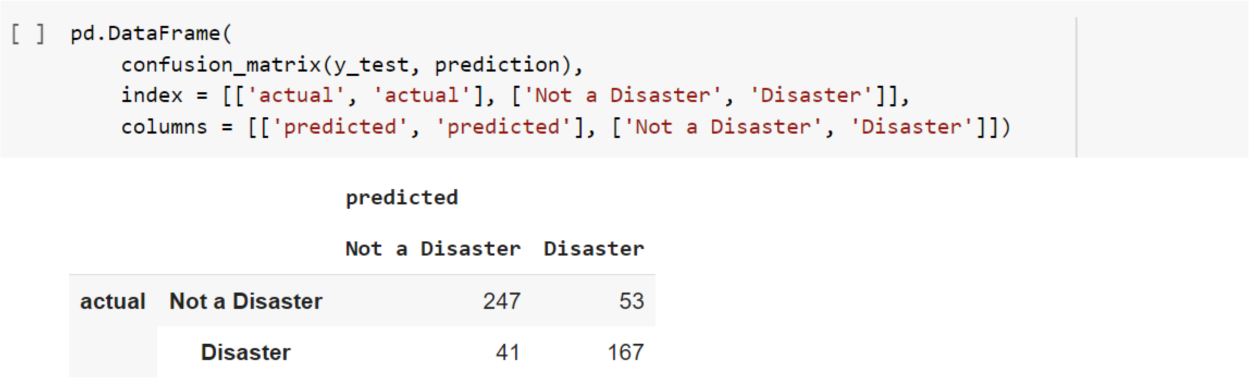
```
[ ]  cls.fit(X_train, y_train)

     SVC(kernel='linear')
```

Model Testing on testing dataset:



```
[ ]  prediction = cls.predict(X_test)
```

```
[ ]  from sklearn.metrics import accuracy_score, confusion_matrix
     print("LinearSVC Accuracy Score is ",accuracy_score(prediction, y_test)*100)

     LinearSVC Accuracy Score is  81.49606299212599
```



```
[ ]  pd.DataFrame(
         confusion_matrix(y_test, prediction),
         index = [['actual', 'actual'], ['Not a Disaster', 'Disaster']],
         columns = [['predicted', 'predicted'], ['Not a Disaster', 'Disaster']])
```

		predicted	
		Not a Disaster	Disaster
actual	Not a Disaster	247	53
	Disaster	41	167

Accuracy achieved: 81.50%