In [1]:
```
import nltk
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.tokenize import RegexpTokenizer
from nltk.stem import WordNetLemmatizer
from nltk.stem import PorterStemmer
from sklearn.feature_extraction.text import TfidfVectorizer
```

In [2]:
```
pip install nltk
```

```
Requirement already satisfied: nltk in d:\vidya\dell\anaconda3\lib\site-packages (3.
7)
Requirement already satisfied: joblib in d:\vidya\dell\anaconda3\lib\site-packages (f
rom nltk) (1.1.0)
Requirement already satisfied: tqdm in d:\vidya\dell\anaconda3\lib\site-packages (fro
m nltk) (4.64.1)
Requirement already satisfied: regex>=2021.8.3 in d:\vidya\dell\anaconda3\lib\site-pa
ckages (from nltk) (2022.7.9)
Requirement already satisfied: click in d:\vidya\dell\anaconda3\lib\site-packages (fr
om nltk) (8.0.4)
Requirement already satisfied: colorama in d:\vidya\dell\anaconda3\lib\site-packages
(from click->nltk) (0.4.5)
Note: you may need to restart the kernel to use updated packages.
```

In [3]:
```
train_data=pd.read_csv("C:\\Users\\DELL\\Downloads\\train nlp.csv")
test_data=pd.read_csv("C:\\Users\\DELL\\Downloads\\test nlp.csv")
```

In [4]:
```
train_data.head()
```

Out[4]:

	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

In [5]:
```
#checking description and info of train
```

In [6]:
```
train_data_description=train_data.describe()
print(train_data_description)
```

```
                 id      target
count   7613.000000  7613.00000
mean    5441.934848     0.42966
std     3137.116090     0.49506
min        1.000000     0.00000
25%     2734.000000     0.00000
50%     5408.000000     0.00000
75%     8146.000000     1.00000
max    10873.000000     1.00000
```

In [7]:
```
train_data_info=train_data.info()
print(train_data_info)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7613 entries, 0 to 7612
Data columns (total 5 columns):
 #   Column    Non-Null Count  Dtype 
---  ------    --------------  ----- 
 0   id        7613 non-null   int64 
 1   keyword   7552 non-null   object
 2   location  5080 non-null   object
 3   text      7613 non-null   object
 4   target    7613 non-null   int64 
dtypes: int64(2), object(3)
memory usage: 297.5+ KB
None
```

In [8]:
```
#removing insignificant columns
train_data=train_data.drop(columns=['keyword','location'])
```

In [9]:
```
train_data.head()
```

Out[9]:

	id	text	target
0	1	Our Deeds are the Reason of this #earthquake M...	1
1	4	Forest fire near La Ronge Sask. Canada	1
2	5	All residents asked to 'shelter in place' are ...	1
3	6	13,000 people receive #wildfires evacuation or...	1
4	7	Just got sent this photo from Ruby #Alaska as ...	1

In [10]:
```
#using nltk tools to preprocess text data
import re
import nltk
from nltk.stem import PorterStemmer
def clean(text):
    pattern=re.compile('[^a-zA-Z]')
    words=nltk.word_tokenize(text)
    stop_words=set(nltk.corpus.stopwords.words('english'))
    words=[PorterStemmer().stem(word)for word in words if word.lower()not in stop_word
    cleaned_text=''.join(words)
    return cleaned_text
```

In [11]:
```
nltk.download()
```

```
showing info https://raw.githubusercontent.com/nltk/nltk_data/gh-pages/index.xml
```

Out[11]:
```
True
```

In [12]:
```
train_data['text_cleaned']=train_data['text'].apply(clean)
```

In [13]:
```
x=train_data['text_cleaned'].values
y=train_data['target'].values
```

In [14]:
```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics import accuracy_score,classification_report,confusion_matrix
```

In [15]:
```
classifier=TfidfVectorizer()
x=classifier.fit_transform(x)
```

In [16]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,random_state=44,test_size=10,strati
```

In [17]:
```
from sklearn.linear_model import LogisticRegression
logreg=LogisticRegression(penalty='l2')
logreg.fit(x_train,y_train)
pred=logreg.predict(x_test)
```

In [18]:
```
R=logreg.predict(x_train)
accuracy_score(y_train,R)
```

Out[18]:
```
0.8559779034591608
```

In [19]:
```
accuracy_score(y_test,pred)
```

Out[19]:
```
0.9
```

In [20]:
```
from sklearn.svm import SVC
model=SVC()
model.fit(x_train,y_train)
```

Out[20]:
```
SVC()
```

In [21]:
```
y_pre=model.predict(x_test)
```

In [22]:
```
score=accuracy_score(y_test,y_pre,normalize=True)
print(score)
```

```
0.9
```

In [23]:
```
test_data['text']=test_data['text'].apply(clean)
```

In [24]:
```
x=classifier.transform(test_data['text'])
```

In [25]:
```
predicts=model.predict(x)
```

In [26]:
```
submission_3=pd.DataFrame({'id':test_data['id'],'target':predicts})
```

In [ ]:
```

```