jupyter SVC_NLP (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted mlenv

Run Code

In [5]:
```
import sklearn
import numpy as np
import pandas as pd
```

In [6]:
```
train = pd.read_csv("Resources/train.csv")
test = pd.read_csv("Resources/test.csv")
sample_submission = pd.read_csv("Resources/sample_submission.csv", dtype=ob
```

In [7]:
```
# install tweet-preprocessor to clean tweets
!pip install tweet-preprocessor
```

```
Requirement already satisfied: tweet-preprocessor in /Users/michaelkori/o
pt/anaconda3/lib/python3.8/site-packages (0.6.0)
```

In [8]:
```
# Remove special character using the regular expression library
import re

# Set up punctuations we want to be replaced
REPLACE_NO_SPACE = re.compile("(\.)|(\;)|(\:)|(\!)|(\')|(\?)|(\,)|(\")|(\|
REPLACE_WITH_SPACE = re.compile("(<br\s/><br\s/?)|(-)|(/)|(:).")
```

In [9]:
```
import preprocessor as p

def clean_tweets(df):
    tempArr = []
    for line in df:
        # Send to tweet_processor
        tmpL = p.clean(line)
        # Remove Punctuation
        tmpL = REPLACE_NO_SPACE.sub("", tmpL.lower()) # Convert All Tweets
        tmpL = REPLACE_WITH_SPACE.sub(" ", tmpL)
        tempArr.append(tmpL)
    return tempArr
```

In [10]:
```
# Clean Training Data
train_tweet = clean_tweets(train['text'])
train_tweet = pd.DataFrame(train_tweet)
```

In [11]:
```
# Append cleaned tweets to the trained data
train['clean_tweet'] = train_tweet

# Compare the cleaned and uncleaned tweets
train.head()
```

Out[11]:

| | id | keyword | location | text | target | clean_tweet |
|---|---|---|---|---|---|---|
| 0 | 1 | NaN | NaN | Our Deeds are the Reason of this #earthquake M... | 1 | our deeds are the reason of this may allah for... |
| 1 | 4 | NaN | NaN | Forest fire near La Ronge Sask. | 1 | forest fire near la ronge sask |

| | | | | | | |
|---|---|---|---|---|---|---|
| | 4 | NaN | NaN | Canada | | canada |
| **2** | 5 | NaN | NaN | All residents asked to 'shelter in place' are ... | 1 | all residents asked to shelter in place are be... |
| **3** | 6 | NaN | NaN | 13,000 people receive #wildfires evacuation or... | 1 | people receive evacuation orders in california |
| **4** | 7 | NaN | NaN | Just got sent this photo from Ruby #Alaska as ... | 1 | just got sent this photo from ruby as smoke fr... |

In [12]:
```
# Drop unnecessary columns

train = train.drop(['text'], axis=1)

train.head()
```

Out[12]:

| | id | keyword | location | target | clean_tweet |
|---|---|---|---|---|---|
| **0** | 1 | NaN | NaN | 1 | our deeds are the reason of this may allah for... |
| **1** | 4 | NaN | NaN | 1 | forest fire near la ronge sask canada |
| **2** | 5 | NaN | NaN | 1 | all residents asked to shelter in place are be... |
| **3** | 6 | NaN | NaN | 1 | people receive evacuation orders in california |
| **4** | 7 | NaN | NaN | 1 | just got sent this photo from ruby as smoke fr... |

In [13]:
```
# Clean the test data and append the cleaned tweets to the test data

test_tweet = clean_tweets(test['text'])
test_tweet = pd.DataFrame(test_tweet)

# Append cleaned tweets to the training data

test['clean_tweet'] = test_tweet

# Compare the cleaned and uncleaned tweets

test.tail()
```

Out[13]:

| | id | keyword | location | text | clean_tweet |
|---|---|---|---|---|---|
| **3258** | 10861 | NaN | NaN | EARTHQUAKE SAFETY LOS ANGELES ÛÒ SAFETY FASTE... | earthquake safety los angeles safety fasteners... |
| **3259** | 10865 | NaN | NaN | Storm in RI worse than last hurricane. My city... | storm in ri worse than last hurricane my city&... |
| **3260** | 10868 | NaN | NaN | Green Line derailment in Chicago http://t.co/U... | green line derailment in chicago |
| **3261** | 10874 | NaN | NaN | MEG issues Hazardous Weather Outlook (HWO) htt... | meg issues hazardous weather outlook hwo |
| **3262** | 10875 | NaN | NaN | #CityofCalgary has activated its Municipal Eme... | has activated its municipal emergency plan |

In [14]:
```
# Drop unnecessary columns

test = test.drop(['text', 'location', 'keyword'], axis=1)

test.head()
```

Out[14]:

| | id | clean_tweet |
|---|---|---|

| | | |
|---|---|---|
| **0** | 0 | just happened a terrible car crash |
| **1** | 2 | heard about is different cities stay safe ever... |
| **2** | 3 | there is a forest fire at spot pond geese are ... |
| **3** | 9 | apocalypse lighting |
| **4** | 11 | typhoon soudelor kills in china and taiwan |

In [15]:
```
# Test and Train split
from sklearn.model_selection import train_test_split

# Extract the labels from the train data

y = train.target.values

# Use 70% for the training and 30% for the test

x_train, x_test, y_train, y_test = train_test_split(train.clean_tweet, y, s
```

In [16]:
```
# Vectorize tweets using CountVectorizer

from sklearn.feature_extraction.text import CountVectorizer
```

In [17]:
```
# Vectorize tweets for model building

vectorizer = CountVectorizer(binary=True)

# Learn a vocabulary dictionary of all tokens in the raw documennts

vectorizer.fit(list(x_train) + list(x_test))

# Transform documents to document-term matrix

x_train_vec = vectorizer.transform(x_train)
x_test_vec = vectorizer.transform(x_test)
```

In [18]:
```
# Apply Support Vector Classifier (SVC)

from sklearn import svm

# Classify using support vector classifier

svm = svm.SVC(kernel = 'linear', probability=True)

# Fit the SVC model based on the given training data

prob = svm.fit(x_train_vec, y_train).predict_proba(x_test_vec)

# Perform classification and prediction on samples in x_test

y_pred_svm = svm.predict(x_test_vec)
```

In [19]:
```
# Accuracy score for SVC
from sklearn.metrics import accuracy_score
```

```
print("Accuracy score for SVC is ", accuracy_score(y_test, y_pred_svm) * 1(
```

```
Accuracy score for SVC is  75.56917688266199 %
```

In [20]:
```
# Calculate f1 Score

from sklearn import feature_extraction, linear_model, model_selection, prep

scores2 = sklearn.metrics.f1_score(y_test, y_pred_svm, average='binary')
```

In [21]:
```
scores2
```

Out[21]:
```
0.704135737009544
```

In [35]:
```
sample_pred_svm = svm.predict(x_train_vec)
mylist = list(sample_pred_svm)
```

In [36]:
```
sample_submission
```

Out[36]:

| | id | target |
|---|---|---|
| **0** | 0 | 0 |
| **1** | 2 | 0 |
| **2** | 3 | 0 |
| **3** | 9 | 0 |
| **4** | 11 | 0 |
| **...** | ... | ... |
| **3258** | 10861 | 0 |
| **3259** | 10865 | 0 |
| **3260** | 10868 | 0 |
| **3261** | 10874 | 0 |
| **3262** | 10875 | 0 |

3263 rows × 2 columns

In [38]:
```
mylist.count(0)
```

Out[38]:
```
3101
```

In [ ]:
```

```