In [ ]:
```
# Importing important libraries

import numpy as np
import pandas as pd
import re # library for using regular expressions
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer # from stemming a word - reducting wo
from sklearn.feature_extraction.text import TfidfVectorizer # converting text da
import nltk
nltk.download('stopwords')
print('All important libraries are imported successfully')
```

In [4]:
```
new_data = "@switchfoot http://twitpic.com/2y1zl - Awww, that's a bummer.  You s
```

In [6]:
```
#Preprocessing and Stemming the Text
portStem = PorterStemmer()

def stemming(content):
    stemmed_content = re.sub('[^a-zA-Z]',' ', content)
    stemmed_content = stemmed_content.lower()
    stemmed_content = stemmed_content.split()
    stemmed_content = [portStem.stem(word) for word in stemmed_content if not wo
    stemmed_content = ' '.join(stemmed_content)
    return stemmed_content
```

In [14]:
```
processed_tweet = [stemming(new_data)]
print(processed_tweet)
```

```
['switchfoot http twitpic com zl awww bummer shoulda got david carr third day']
```

In [17]:
```
import pickle
```

In [24]:
```
# Load the saved vectorizer
with open('trained_vectorizer.pkl', 'rb') as f:
    vectorizer = pickle.load(f)  # Load the vectorizer
```

In [25]:
```
# Transform the new data using the loaded vectorizer
vectorized_tweet = vectorizer.transform(processed_tweet)
```

In [27]:
```
# loading the saved model
loaded_model = pickle.load(open('twitter_sentiment_analysis_model.sav', 'rb'))
```

In [29]:
```
prediction = loaded_model.predict(vectorized_tweet)
print(prediction)

if (prediction[0] == 0):
    print("Negative Tweet")
else:
    print("Positive Tweet")
```

```
[0]
Negative Tweet
```