الغاية من الجلسة:

التعرف على خطوات وأدوات المعالجة المسبقة للنص باستخدام مكتبة NLTK المعرفة ضمن لغة Python

مقدمة:

معالجة النص تعتبر من أهم الخطوات ضمن عملية تحليل النص والتعامل مع النصوص عامة

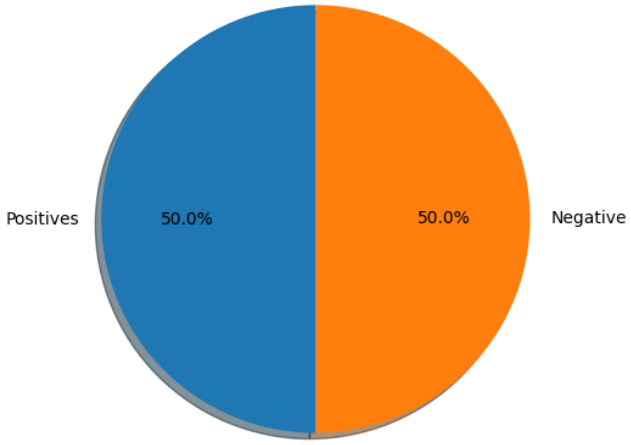
في هذه الجلسة سنقوم بتطبيق عدة عمليات وهي:

- تحويل الأحرف الى أحرف صغيرة
- إزالة علامات الترقيم
- إزالة كلمات التوقف
- إزالة Hyper link وغيرها
- إزالة emojis

```
import nltk # Python library for NLP
from nltk.corpus import twitter_samples # sample Twitter dataset from NLTKimport
matplotlib.pyplot as plt # library for visualization
import random # pseudo-random number generator
```

في البداية تم استدعاء المكتبات اللازمة ومن ثم قراءة ملف مجموعة البيانات والذي يحتوي على 5000 tweets إيجابي و 5000 tweets سلبي

```
nltk.download('twitter_samples')
# select the set of positive and negative Tweets
all_positive_tweets = twitter_samples.strings('positive_tweets.json')
all_negative_tweets = twitter_samples.strings('negative_tweets.json')
```




```
# download the stopwords from NLTK
nltk.download('stopwords')
```

المعني بتحويل الأحرف الى أحرف صغيرة()lowerمن الممكن تطبيق التابع

سنقوم الآن بإزالة كلمات التوقف وعلامات الترقيم.

كلمات التوقف هي كلمات توجد في الجملة ولكن إزالتها لا تؤثر على معنى أو مفهوم الجملة لذلك من الممكن أن تعيق تلك الكلمات عملية فهم الآلة للنص، أمثلة على كلمات التوقف في اللغة الإنكليزية.

'a', 'about', 'above', 'after', 'again', 'against', 'ain', 'all', 'am', 'an', 'and', 'any', 'are', 'aren', "aren't"

```
tweets_clean = []
for word in tweet_tokens: # Go through every word in your tokens list
 if (word not in stopwords_english and  word not in string.punctuation): # remove punctuation
tweets_clean.append(word)
print('removed stop words and punctuation:')print(tweets_clean)
```

['my', 'beautiful', 'sunflowers', 'on', 'a', 'sunny', 'friday', 'morning', 'off', ':)', 'sunflowers', 'favourites', 'ha

removed stop words and punctuation:

['beautiful', 'sunflowers', 'sunny', 'friday', 'morning', ':)', 'sunflowers', 'favourites', 'happy', 'friday', '…']

إزالة الرموز الموجودة في النص:

```
print('\033[92m' + tweet)

print('\033[94m')# remove old style retweet text "RT"

tweet2 = re.sub(r'^RT[\s]+', '', tweet)# remove hyperlinks

tweet2 = re.sub(r'https?:\/\/.*[\r\n]*', '', tweet2)# remove hashtags# only removing the hash # sign from the word

tweet2 = re.sub(r'#', '', tweet2)print(tweet2)
```

My beautiful sunflowers on a sunny Friday morning off :) #sunflowers #favourites #happy #Friday off… https://t.co/3tfYo

My beautiful sunflowers on a sunny Friday morning off :) sunflowers favourites happy Friday off…

والتي من الممكن أن تكون Token والمقصود به هو عملية تقسيم النص إلى Tokenizationالمرحلة التالية هي مرحلة Tokenكلمة أو جملة ، في المثال التالي تم تقسيم النص إلى كلمات وكل كلمة تعتبر

```
# instantiate tokenizer Class

tokenizer = TweetTokenizer(preserve_case=False, strip_handles=True, reduce_len=True)# tokenize tweets

tweet_tokens = tokenizer.tokenize(tweet2)

print()

print('Tokenized string:')

print(tweet_tokens)
```

My beautiful sunflowers on a sunny Friday morning off :) sunflowers favourites happy Friday off…
Tokenized string:

['my', 'beautiful', 'sunflowers', 'on', 'a', 'sunny', 'friday', 'morning', 'off', ':)', 'sunflowers', 'favourites', 'ha ……

Instantiate stemming class

```
stemmer = PorterStemmer()# Create an empty list to store the stems
tweets_stem = []
for word in tweets_clean:
 stem_word = stemmer.stem(word) # stemming word
 tweets_stem.append(stem_word) # append to the list
print('stemmed words:')
print(tweets_stem)
```

porterstemmerهي أداة تستخدم لإعادة الكلمة الى جذرها بمعنى إزالة المحارف المضافة والتي لا تؤثر على معنى الكلمة.

['beautiful', 'sunflowers', 'sunny', 'friday', 'morning', ':)', 'sunflowers', 'favourites', 'happy', 'friday', '…']

stemmed words:

['beauti', 'sunflow', 'sunni', 'friday', 'morn', ':)', 'sunflow', 'favourit', 'happi', 'friday', '…']

إزالة emoji

```
def remove_emoji(string):
emoji_pattern = re.compile("["
u"\U0001F600-\U0001F64F" # emoticons
u"\U0001F300-\U0001F5FF" # symbols & pictographs
u"\U0001F680-\U0001F6FF" # transport & map symbols
u"\U0001F1E0-\U0001F1FF" # flags (iOS)
u"\U00002702-\U000027B0"
u"\U000024C2-\U0001F251"
"]+", flags=re.UNICODE)
return emoji_pattern.sub(r'', string)
'game is on '
remove_emoji("game is on 🔥🔥")
```

من الممكن أن يكون النص المدخل يحتوي على أخطاء إملائية من الممكن إزالة تلك الأخطاء

```
from spellchecker import SpellChecker

spell = SpellChecker()

def correct_spellings(text):

corrected_text = []

misspelled_words = spell.unknown(text.split())

for word in text.split():

if word in misspelled_words:

 corrected_text.append(spell.correction(word))

else:

 corrected_text.append(word) return " ".join(corrected_text)text = "speling correctin"

print('The text before correcting: '+ text)print('The text after correcting'+ correct_spellings(text))
```