Data Pre-Processing Steps___Removing Punctuations

By Venkatesh_Mungi venkateshmungi1247@gmail.com

In [1]:
```
# To show the punctuations
import string
string.punctuation
```

Out[1]:
```
'!"#$%&\'()*+,-./:;<=>?@[\\]^_`{|}~'
```

In [2]:
```
# Removing the Punctuations from given text
regular_punctuations = list(string.punctuation)
def remove_punctuation(text,punct_list):
    for punct in punct_list:
        if punct in text:
            text = text.replace(punct, " ")
    return text.strip()

remove_punctuation("Hi!. How are you ??", regular_punctuations)
```

Out[2]:
```
'Hi   How are you'
```

In [3]:
```
reg_pun = list(string.punctuation)

def rem_punc(text, punc_list):
    for pun in punc_list:
        if pun in text:
            text = text.replace(pun, " ")
    return text.strip()

rem_punc("Good morning!, Where are you Now?", reg_pun)
```

Out[3]:
```
'Good morning   Where are you Now'
```

In [4]:
```
import pandas as pd
import numpy as np

df = pd.read_csv(r"C://PYTHON//AI_ML//NLP//covid19_tweets.csv")
df.head(3)
```

Out[4]:

	user_name	user_location	user_description	user_created	user_followers	user_friends	user_favourites	user_verified	date	text	hashtags	source	is_retweet
0	ᏉᎥ☻լℭ↑	astroworld	wednesday addams as a disney princess keepin i...	2017-05-26 05:46:42	624	950	18775	False	2020-07-25 12:27:21	If I smelled the scent of hand sanitizers toda...	NaN	Twitter for iPhone	False
1	Tom Basile 🇺🇸	New York, NY	Husband, Father, Columnist & Commentator. Auth...	2009-04-16 20:06:23	2253	1677	24	True	2020-07-25 12:27:17	Hey @Yankees @YankeesPR and @MLB - wouldn't it...	NaN	Twitter for Android	False
2	Time4fisticuffs	Pewee Valley, KY	#Christian #Catholic #Conservative #Reagan #Re...	2009-02-28 18:57:41	9275	9525	7254	False	2020-07-25 12:27:14	@diane3443 @wdunlap @realDonaldTrump Trump nev...	['COVID19']	Twitter for Android	False

1. Remove punctuation by using regex

The regex package in python allows us to manage, control, and manipulate regular expressions of various types.

In [5]:
```
import re
```

In [6]:
```
strings = "@$%Ratna&*Priya#! is a,good girl?/.!@#$%^&*()_+"
new_string = re.sub(r"[^\w\s]",'',strings)
print("strings:",strings)
print("String without Punctuation: ",new_string)
```

```
strings: @$%Ratna&*Priya#! is a,good girl?/.!@#$%^&*()_+
String without Punctuation:  RatnaPriya is agood girl_
```

Removing Underscores

Removing leading underscores from a string

You can use the string lstrip() function to remove leading underscores from a string in Python. The lstrip() function is used to remove characters from the start of the string and by default removes leading whitespace characters.

To remove leading underscores with the lstrip() function, pass the underscore character, '_' as an argument.

Note that the lstrip() function only removes characters from the start of the string.

In [7]:
```
beeshma = "_Great warrior of the mahabharat"
beeshma.lstrip("_")
```

Out[7]:
```
'Great warrior of the mahabharat'
```

Remove all underscores

To remove all the occurrences of the underscore character irrespective of where it occurs in the string, you can use the string replace() function.

In [8]:
```
arjuna = "He_is_the only warrior and who_can_beat beeshma"
arjuna.replace("_","")
```

Out[8]:
```
'Heisthe only warrior and whocanbeat beeshma'
```

In [9]:
```
str = '52_841_63_24_76_49'
#split string by _
items = str.split('_')
items
```

Out[9]:
```
['52', '841', '63', '24', '76', '49']
```

2. Remove punctuation from string by using the translate method

In python, the string function is the quickest way for punctuation removal. To utilize the translate function, we must first import the string module. Let me explain if we are unsure what the translate function does. The translate method produces a string in which some characters are substituted with characters from a dictionary or a mapping table.The example below shows removing punctuation from strings using the translate method.

In [10]:
```
import string
strin = "nltk @%,^ remove #! punctualtion"
rm_strin = strin.translate (str.maketrans ('', '', string.punctuation))
print ('String: ', strin)
print ('Without punctuation string: ', rm_strin)
```

```
String:  nltk @%,^ remove #! punctualtion
Without punctuation string:  nltk  remove  punctualtion
```

3. Remove the punctuation by using the join method

The join method can also be used to remove the punctuation. If we are unfamiliar with the join approach, let me describe it shortly. The join method allows us to create s trings from iterable objects in various ways. It concatenates each component of an iterable.The below example shows removing punctuation by using the join method.

In [11]:
```
import string
py_str = "nltk @%,^ remove #! punctualtion"
exclude = set(string.punctuation)
py_str = ''.join(ch for ch in py_str if ch not in exclude)
print (py_str)
```

```
nltk  remove  punctualtion
```

4.Remove the punctuation by using replace method

Replace method is a quick and easy technique to remove punctuation. It gives us an object to the iterator.Many helpful techniques are available in Python strings. Replace is an example of such a procedure.We can use this method to replace one substring of characters in a string with another. This function default clears the string of all p articular characters or substring occurrences.We may limit the occurrences by supplying a count value to the replace method as the third parameter.The below example shows that removing the punctuation using the remove method is as follows.

In [12]:
```
py_str = "nltk @emove punctuation"
print (py_str)
py_ls = py_str.replace ('@', 'r')
print (py_ls)
```

```
nltk @emove punctuation
nltk remove punctuation
```

Removing punctuations in dataframe using for loop

In [13]:
```
data = {"A":["Orange@","Pink.","Yell#ow"],
        "B":["Dad's","Mom;s","Nan"],
        "C":["Intrested.","Not-intrested","nop@e"],
        "D":["Navy","NaN","NaN"]}

frame = pd.DataFrame(data, index =[0,1,2])

frame
```

Out[13]:

	A	B	C	D
0	Orange@	Dad's	Intrested.	Navy
1	Pink.	Mom;s	Not-intrested	NaN
2	Yell#ow	Nan	nop@e	NaN

In [14]:
```
# “remove punctuation in dataframe column”

# Define the function to remove the punctuation.

def remove_punctuations(text):
    for punctuation in string. punctuation:
        text = text. replace(punctuation, '')
    return text
    # Apply to the DF series.

frame['C'] = frame['C'].apply(remove_punctuations)
frame['B'] = frame['B'].apply(remove_punctuations)
frame['D'] = frame['D'].apply(remove_punctuations)
frame['A'] = frame['A'].apply(remove_punctuations)
frame
```

Out[14]:

	A	B	C	D
0	Orange	Dads	Intrested	Navy
1	Pink	Moms	Notintrested	NaN
2	Yellow	Nan	nope	NaN

In [15]:
```
frame['D'] = frame['D'].str.replace('NaN', '')
```

In [16]:
```
frame
```

Out[16]:

	A	B	C	D
0	Orange	Dads	Intrested	Navy
1	Pink	Moms	Notintrested	
2	Yellow	Nan	nope	

In [17]:
```
punctuation= '''!()-[]{};:'"\, <>./?@#$%^&*_~'''
print("The punctuation marks are: ", punctuation)
myString= "Python.:F}or{Beg~inn;ers"
print("Input String is: ", myString)
newString=""
for x in myString:
    if x not in punctuation:
        newString=newString+x
print("Output String is: ", newString)
```

```
The punctuation marks are:  !()-[]{};:'"\, <>./?@#$%^&*_~
Input String is:  Python.:F}or{Beg~inn;ers
Output String is:  PythonForBeginners
```