Natural Language processing

TP 3 – Term Frequency - Inverse Document Frequency (TF – IDF)

LE Do Thanh Dat, YOU Borachhun

Exercice 1 - BOW

In [2]:
```
# 1. Importer les dépendances
## import statements ##
import numpy as numpy
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import re

from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.linear_model import LogisticRegression
from sklearn.pipeline import Pipeline
from sklearn.metrics import confusion_matrix, classification_report, accuracy_sc

sns.set_context('talk')
sns.set_color_codes()
plot_kwds = {'alpha' : 0.25, 's' : 80, 'linewidths' : 0}

import warnings; warnings.simplefilter('ignore')
```

In [3]:
```
# 2. Importer les données et déclarer quelques variables
data_files = './data/Comment Spam.xls'

data = pd.read_excel(data_files)
data = data[['Comment', 'Class']]
train_data = data
train_data.head()
```

Out[3]:

	Comment	Class
0	this song is racist	0
1	and how many subscribers compared to her over ...	1
2	HI! CHECK OUT OUR AWESOME COVERS! AND SAY WHAT...	1
3	well done shakira	0
4	:D subscribe to me for daily vines	1

In [5]:
```
# 3. Prétraitement des données
def process_content(content):
    return " ".join(re.findall("[A-Za-z]+", content.lower()))

train_data['processed_comment'] = train_data['Comment'].apply(process_content)
```

```
In [6]: # 4. Observer l’avant et l’après prétraitement
        train_data.head(20)
```

Out[6]:

	Comment	Class	processed_comment
0	this song is racist	0	this song is racist
1	and how many subscribers compared to her over ...	1	and how many subscribers compared to her over ...
2	HI! CHECK OUT OUR AWESOME COVERS! AND SAY WHAT...	1	hi check out our awesome covers and say what y...
3	well done shakira	0	well done shakira
4	:D subscribe to me for daily vines	1	d subscribe to me for daily vines
5	Part 2. Holy Mary, pray for us Holy Mother of ...	1	part holy mary pray for us holy mother of god ...
6	I really can&#39;t comprehend Miley Cyrus , s...	1	i really can t comprehend miley cyrus she actu...
7	Nice song ^_^	0	nice song
8	This makes me miss the world cup	0	this makes me miss the world cup
9	******* Facebook is LAME and so 2004! Check ou...	1	facebook is lame and so check out swagfriends ...
10	I hope everyone is in good spirits I&#39;m a h...	1	i hope everyone is in good spirits i m a hard ...
11	:)	0	
12	She is good	0	she is good
13	Subscribe to my Youtube Channel!! :) Suscribit...	1	subscribe to my youtube channel suscribite a m...
14	beautiful	0	beautiful
15	Earn money for being online with 0 efforts! ...	1	earn money for being online with efforts bit l...
16	**CHECK OUT MY NEW MIXTAPE**** **CHECK OUT MY ...	1	check out my new mixtape check out my new mixt...
17	Hello everyone, It Is not my intention to spam...	1	hello everyone it is not my intention to spam ...
18	******* Facebook is LAME and so 2004! Check ou...	1	facebook is lame and so check out swagfriends ...
19	Could you please check out my covers on my cha...	1	could you please check out my covers on my cha...

```
In [7]: #  Train, test split
        X_train, X_test, y_train, y_test = train_test_split(train_data['processed_commer
                                                            train_data['Class'], test_si
```

```
In [8]: # 6
        model = Pipeline([('vect', CountVectorizer(stop_words='english')),
                          ('tfidf', TfidfTransformer()),
                          ('clf', LogisticRegression()),
                          ])
```

In [9]:
```
# 7. Exécuter le pipeline
#Pipeline execution
model.fit(X_train, y_train)

#Prediction with test set
predicted = model.predict(X_test)

#Confusion matrix calculation
confusion_matrix(y_test, predicted)
```

Out[9]:
```
array([[132,   7],
       [  9, 112]], dtype=int64)
```

In [10]:
```
# 8. Imprimer le rapport de performance du modele
print('accuracy_score', accuracy_score(y_test, predicted))
print('Reporting...')
print(classification_report(y_test,predicted))
```

```
accuracy_score 0.9384615384615385
Reporting...
              precision    recall  f1-score   support

           0       0.94      0.95      0.94       139
           1       0.94      0.93      0.93       121

    accuracy                           0.94       260
   macro avg       0.94      0.94      0.94       260
weighted avg       0.94      0.94      0.94       260
```

In [11]:
```
# 9. Imprimer les résultats de la validation croisée
#cross validation on training
print(cross_val_score(model, X_train, y_train, cv=5))

#cross validation on test
print(cross_val_score(model, X_test, y_test, cv = 5))
```

```
[0.94230769 0.95192308 0.94230769 0.9375     0.90865385]
[0.92307692 0.94230769 0.94230769 0.84615385 0.94230769]
```

In [15]:
```
# 10. Essayer quelques phrases afin de les classifier comme spam ou ham
#testing a comment and doing the classification
c1 = ['Im super happy']
content = pd.DataFrame(c1, columns=['Comment'])
content['processed_comment'] = content['Comment'].apply(process_content)
content_processed = content['processed_comment']
#prediction of the class: 0=ham, 1=spam
model.predict(content_processed)
```

Out[15]:
```
array([0], dtype=int64)
```

In [16]:
```
# 11. Faire la prédiction sur les diffèrent exemples
c1 = ['Subscribe to my Youtube Channel!! :)',
      'Hi veronica, hope you are doing good',
      'Earn money for being online with 0 efforts! ...']

#test data pre treatment
```

```
test_data = pd.DataFrame(c1, columns=['Comment'])
test_data['processed_comment'] = test_data['Comment'].apply(process_content)
x_test_new = test_data['processed_comment']

#prediction of the class: 0=ham, 1=spam
model.predict(x_test_new)
```

```
Out[16]: array([1, 0, 1], dtype=int64)
```