Classification des articles de presse

Réalisé par:
- MEHEMEL Souhaib
- ALI Djamel

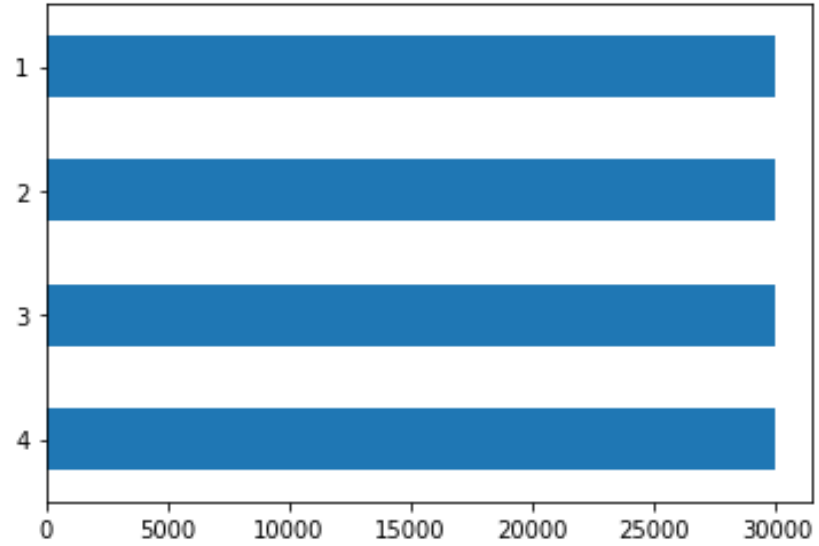
Classification des articles de presse

En 4 catégories :
1. World; 2. Sports; 3. Business; 4. Sci/Tech

Training dataset : 120 000 lignes (4 catégories : 30 000 lignes / catégorie)
Testing dataset : 7 600 lignes (4 catégories : 1 900 lignes / catégorie)

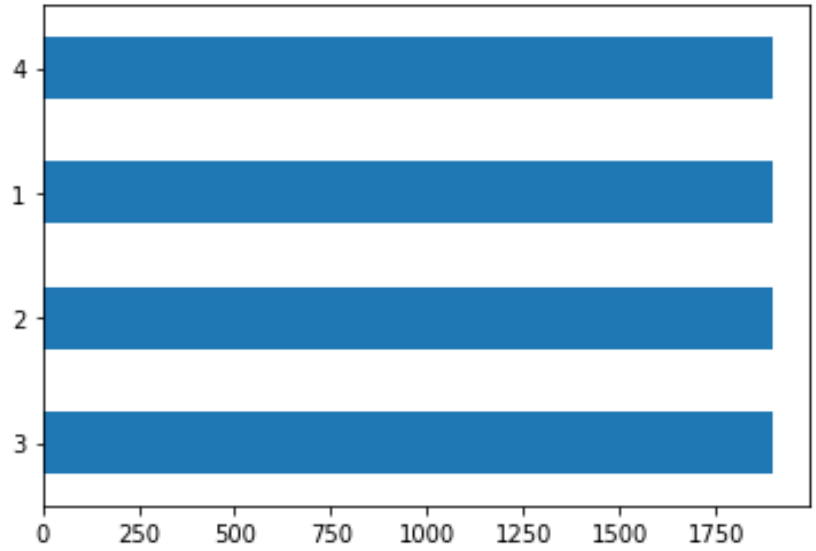
Dataset (trainning set)

Training dataset : 120 000 lignes (4 catégories : 30 000 lignes / catégorie)



Dataset (testing set)

```
Testing dataset  : 7 600 lignes (4 catégories : 1 900 lignes / catégorie)
```



Dataset

Source: https://www.kaggle.com/amananandrai/ag-news-classification-dataset?select=train.csv

Les caractéristiques du dataset sont:

Nombre d'attributs : 3 (*Class Index, Titre, Description*)
Information sur les attributs : catégorie des articles de presse (*world, sport, business, sci-tech*) codée par des entiers de 1 à 4.
Valeurs d'attribut manquantes : Aucune
Créateur : *Aman Anand*
Date : 20 avril 2020

Nettoyage

Étapes clés:

- Le dataset se compose de 2 fichiers csv (train et test)
- S’assurer qu’il n’y a pas de valeurs manquantes (NA value)
- Préprocessing du texte (“tokenization”, supprimer les “mots vides” anglais, “Lemmatization” avec WordNetLemmatizer).
- D’autres essais de pré-processing qui n’ont pas augmenté l’accuracy comme :
  * Suppression de la ponctuation (? , . ! ; : ...)
  * Suppression des mots d'au plus 2 lettres
  * Suppression des nombres
  * Passer tout le contenu en minuscule

```python
# Text preprocessing function
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer

import re

def preprocess(text):

  # word tokenization
  tokens = word_tokenize(text)

  # Remove stop words
  stop_words = stopwords.words("english")
  words = [word for word in tokens if word not in stop_words]

  # lemmatization
  lemmatizer = WordNetLemmatizer()
  lemmatized_words = [lemmatizer.lemmatize(word) for word in words]

  return " ".join(lemmatized_words)
```

Feature Extraction

- Ajout de la colonne « **Preprocessed_text** » qui contient le résultat du preprocessing de ***titre*** + ***description*** de chaque article de presse (une entée = un article de presse).
- Partie « Feature Extraction » : les catégories **1-2-3-4** deviennent respectivement '***World News***', '***Sports News***', '***Business News***', '***Science-Technology News'*** et la colonne '***Class Index'*** s'appelle désormais ***'category'***
- On passe de 5 colonne à 3 (avec ID).

	Preprocessed_Text	category
0	Wall St. Bears Claw Back Into Black (Reuters ...	Business News
1	Carlyle Looks Toward Commercial Aerospace (Re...	Business News

Variables importantes et aperçu de leurs contenus

```python
# train data
X_train = train_data['Preprocessed_Text']
Y_train = train_data['category']

# test data
X_test = test_data['Preprocessed_Text']
Y_test = test_data['category']
X_train.head(-5)
```

```
0         Wall St. Bears Claw Back Into Black ( Reuters ...
1         Carlyle Looks Toward Commercial Aerospace ( Re...
2         Oil Economy Cloud Stocks ' Outlook ( Reuters )...
3         Iraq Halts Oil Exports Main Southern Pipeline ...
4         Oil price soar all-time record , posing new me...
                              ...
119990    Barack Obama Gets # 36 ; 1.9 Million Book Deal...
119991    Rauffer Beats Favorites Win Downhill VAL GARDE...
119992    Iraqis Face Winter Shivering Candlelight BAGHD...
119993    AU Says Sudan Begins Troop Withdrawal Darfur A...
119994    Syria Redeploys Some Security Forces Lebanon B...
Name: Preprocessed_Text, Length: 119995, dtype: object
```

```python
# train data
X_train = train_data['Preprocessed_Text']
Y_train = train_data['category']

# test data
X_test = test_data['Preprocessed_Text']
Y_test = test_data['category']
Y_train.head(-5)
```

```
0         Business News
1         Business News
2         Business News
3         Business News
4         Business News
              ...
119990       World News
119991      Sports News
119992       World News
119993       World News
119994       World News
Name: category, Length: 119995, dtype: object
```

Encodage

TF-IDF

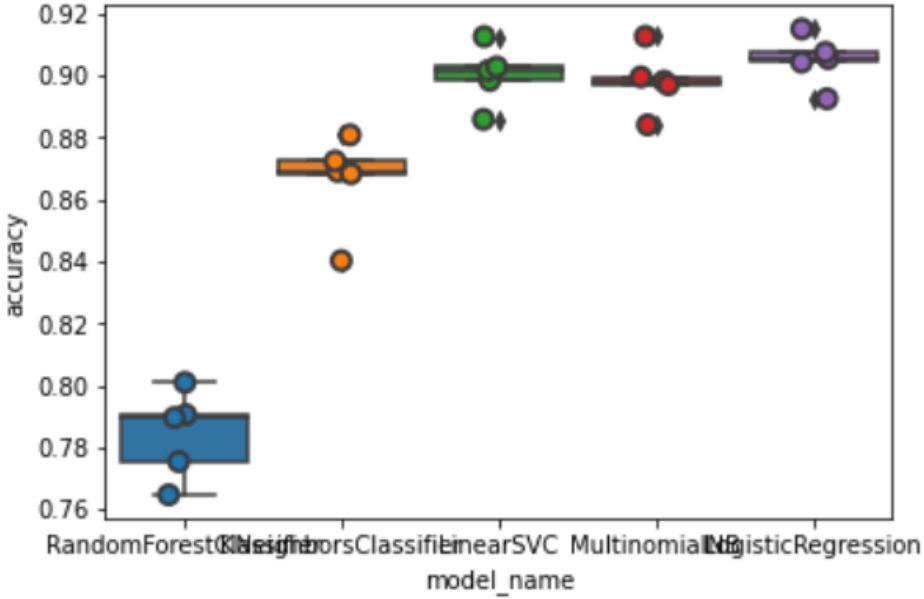
```
# tf-idf
from sklearn.feature_extraction.text import TfidfVectorizer

tf_vec = TfidfVectorizer()
train_features = tf_vec.fit(X_train)
train_features = tf_vec.transform(X_train)
test_features = tf_vec.transform(X_test)
train_features.shape
```

```
(120000, 63030)
```

Choix du Modèle

- Évaluer plusieurs models (utilisés avec ces parametres par defauts)




```
model_name
KNeighborsClassifier      0.866233
LinearSVC                 0.900333
LogisticRegression        0.905058
MultinomialNB             0.898375
RandomForestClassifier    0.784242
Name: accuracy, dtype: float64
```

Logistic regression (Tuning)

- **Trouver de bonnes valeurs aux paramètres de cet algo (solver, max_iter, C, ...).**

- Parametres choisis :
  solver = ‘saga’
  c = 2.1

- Accuracy atteinte = 0.9190789473684211 ≈ **92%**

Logistic regression (résultats)

```
CPU times: user 2 µs, sys: 1 µs, total: 3 µs
Wall time: 11.4 µs

        Used algorithm :  LogisticRegression

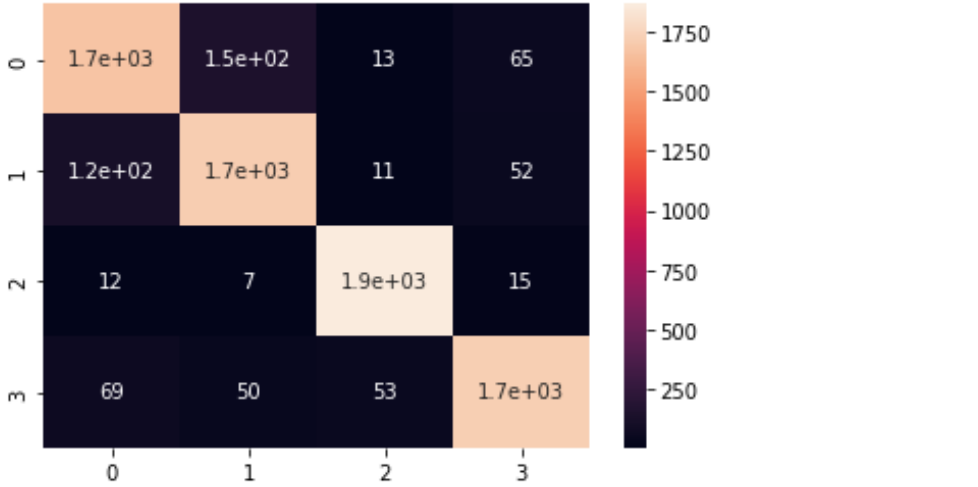

accuracy 0.9190789473684211
              precision    recall  f1-score   support

       World       0.89      0.88      0.89      1900
       Sport       0.89      0.90      0.90      1900
    Business       0.96      0.98      0.97      1900
    Sci-Tech       0.93      0.91      0.92      1900

    accuracy                           0.92      7600
   macro avg       0.92      0.92      0.92      7600
weighted avg       0.92      0.92      0.92      7600
```

```
Used algorithm :  LogisticRegression

<matplotlib.axes._subplots.AxesSubplot at 0x7f30b2f3add0>
```



KNN (tuning)

- Utilisation de la validation croisée pour le réglage des paramètres (parameter tuning, CV = 10 folds).

```
print([i for i in range(1,50,2)])
print(len(k_list))
```

```
[1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, 37, 39, 41, 43, 45, 47, 49]
25
```

- Le paramètre à régler c'est la valeur de k, 25 valeurs différentes de k on été testées (chacune par le principe de la validation croisée), à la fin on choisi la valeur de k qui donne l'erreur de classification la plus petite (c.f. code et graphe de la slide suivante).

KNN (tuning)

- Utilisation de la **cross-validation** pour trouver une bonne valeur de **k**.

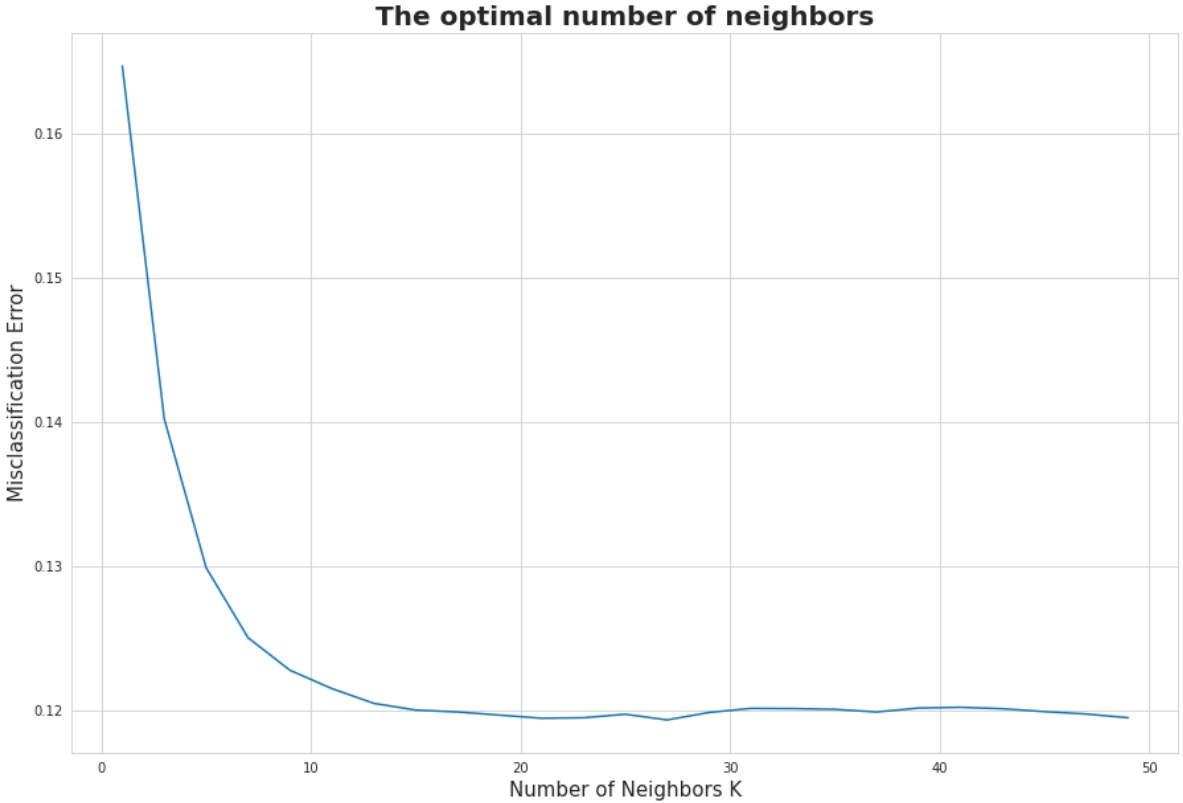
```
# creating list of K for KNN
k_list = list(range(1,50,2))

# creating list of cv scores
cv_scores = []

# perform 10-fold cross validation
for k in k_list:
    knn = KNeighborsClassifier(n_neighbors=k)
    scores = cross_val_score(knn, train_features, Y_train, cv=10, scoring='accuracy')
    cv_scores.append(scores.mean())
```

```
plt.show()
```

```
<Figure size 432x288 with 0 Axes>
```




```
[ ] # finding best k
    best_k = k_list[MSE.index(min(MSE))]
    print("The optimal number of neighbors is %d." % best_k)
```

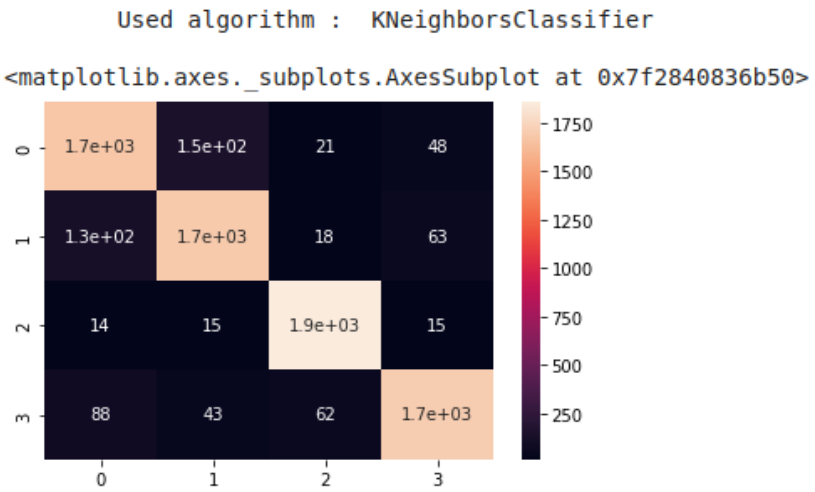
```
The optimal number of neighbors is 27.
```

KNN (résultats)

- Matrice de confusion obtenue.

```
        Used algorithm :  KNeighborsClassifier

array([[1682,  149,   21,   48],
       [ 128, 1691,   18,   63],
       [  14,   15, 1856,   15],
       [  88,   43,   62, 1707]])
```



- Recall (≈95%),
- précision (≈96%)
- et accuray(≈**91%**):

- recall = 6936 / 7286 = 0.9519626681306615
- precision = 6936 / 7250 = 0.9566896551724138

- ```
  accuracy = sum(TP + TN) / sum(TP+FP+FN+TN)
  accuracy =  6936 / 7600  =  0.9126315789473685
  ```
  ```

KNN (résultats)

- Recall, précision et accuray (avec plus de détails sur chaque classe):

```
Used algorithm :  KNeighborsClassifier


accuracy 0.9126315789473685
                          precision    recall  f1-score   support

           Business News       0.88      0.89      0.88      1900
Science-Technology News       0.89      0.89      0.89      1900
            Sports News       0.95      0.98      0.96      1900
             World News       0.93      0.90      0.91      1900

               accuracy                           0.91      7600
              macro avg       0.91      0.91      0.91      7600
           weighted avg       0.91      0.91      0.91      7600
```

Résultats

- N’ayant plus de temps pour essayer toutes les combinaisons et ajustements possibles, notre modèle a atteint jusqu’à présent une exactitude (accuracy) de classement d’environ **92%** (avec le modèle ***Régression logistique***).

- Pareil pour ***KNN*** qui a une accuracy de **91%.**

Discussion

- On pense qu’on a bien fait les toutes premières étapes (récupération des données, pré-processing (en essayant plusieurs cas), TF-IDF).
- Toutefois, on se demande quand même si on ne peut pas faire mieux que ça au niveau du pré-processing.

- On se demande si on peut vraiment aller (très loin) au-delà de l’accuracy = **92%** .
- ...

Code

Parties pertinentes du code déjà vues précédemment.
Lien du nootebook: https://colab.research.google.com/drive/1_BfG3zLWv7d85-Z8mmo_ZO9uU9usxV1C?usp=sharing

Contributions

- **Idée du projet**: Souhaib (au départ, **idée** du *moteur de recommandation d'e-book: Djamel).*
- **Code**:
  - Utilisation de la documentation sklearn (ou directement sur Jupyter Notebook)
  - Aussi, s'inspirer et regarder des exemples publiés un peu partout sur internet, particulièrement sur (c.f. slide suivante pour les détails):
  - *https://www.kaggle.com/*
  - *https://towardsdatascience.com/*
  - *https://medium.com/*
- **Nettoyage des données**: Les 2, (d'abord Souhaib, puis Djamel)
- **Un premier essai avec le modele Naïve Bayes :** Souhaib
- **Modèle selection:** Djamel
- **Algo RegLogistic + KNN**: Les 2
- **Tuning des paramètres:** Les 2
- **Graphes:** Les 2.
- **Slides**: Les 2.

Ressources

*** Documentation sur les étapes du prétraitement du text :**
-> https://www.kaggle.com/anushreepatil01/ag-news-classification
*** Utilisation de la validation croisée pour le choix du paramètre k :**
-> https://www.kaggle.com/skalskip/iris-data-visualization-and-knn-classification?scriptVersionId=1543675&cellId=55

*** Choix du modèle le plus approprié en utilisant la validation croisée:**
-> https://towardsdatascience.com/multi-class-text-classification-with-scikit-learn-12f1e60e0a9f

*** Un exemple de classifieur "régression logisitique":**
-> https://www.kdnuggets.com/2018/11/multi-class-text-classification-model-comparison-selection.html

*** Classieur KNN : (moi-même (Djamel) en regardant la doc)**
et pour chaque modèle, regarder la doc pour mieux comprendre le rôle de chaque paramètre
et lui affecter la valeur la plus convenable et appropriée à notre contexte.


THANK
You