Tut 3: Hướng dẫn tạo TF-IDF và sử dụng KNN

- **Tác giả: Vũ Đức Duy.**
- **Nhóm nghiên cứu về AI của Trường Đại học Bách Khoa Thành phố Hồ Chí Minh.**

1. Khởi chạy python

- Đầu tiên các bạn chạy chương trình Python trong máy (tham khảo Tut 0).

2. Import thư viện

- Đoạn code sau đây sẽ import những thư viện cần thiết:

```
import numpy as np        # Thư viện dùng để xử lý
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.neighbors import KNeighborsClassifier
```

3. Tạo ma trận TF-IDF

- Sau khi các bạn sử dụng crawler để lấy dữ liệu documents về máy thì hãy chia các document này ra từng file riêng lẻ và để trong cùng một folder. Ví dụ mình có 3 documents với file path tương ứng như sau:

  /home/documents/doc1.txt

  /home/documents/doc2.txt

  /home/documents/doc3.txt

- Quay trở lại Python các bạn khởi tạo 1 list chứa file path của tất cả các file document (lưu ý giữa đường dẫn tương đối và đường dẫn tuyệt đối) (Mình đang dùng ubuntu nên đường dẫn được phân cách bằng dấu "/" còn trong windows thì dùng "\").

- Ví dụ giả sử bạn có 100 file và nằm ở thư mục như trên:

  fileList = ["/home/documents/doc" + str(i) + ".txt" for i in range(1,101)]

- Sau đó các bạn sử dụng hàm TfidfVectorizer để tính ma trận TF-IDF

  X = TfidfVectorizer(input="filename").fit_transform(fileList).toarray()

- Chạy lệnh sau để in kết quả:

  print(X)

- Chi tiết về prototype của hàm các bạn có thể tham khảo ở link dưới đây:

  http://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

  http://scikit-learn.org/stable/modules/feature_extraction.html#text-feature-extraction

4. KNN

- Trước khi đưa dữ liệu huấn luyện vào model, chúng ta cần gán label cho từng document, label ở bài lab này là category. Các bạn mã hóa category của từng document rồi gộp lại thành 1 vector có kích thước là (1,n) với n là số lượng document.
- Sau khi đã có ma trận TF-IDF (X) và label vector (tạm gọi là Y), thì có thể bắt đầu đưa vào KNN.
- Chạy lệnh sau:

  n = 5 # Đây là tham số số lượng neighbors của KNN, bạn có thể thử thay bằng 1 số khác nếu muốn

  neigh = KneighborsClassifier(n_neighbors=3)
  neigh.fit(X, Y) # X và Y là training data, tương ứng với TF-IDF và label vector

- Sau khi chạy lệnh thì model đã được "fit" vào tập data training, bây giờ bạn có thể dự đoán được 1 document mới sẽ được phân loại vào category nào thông qua hàm predict:

  neigh.predict([new_X]) # new_X là 1 document mới chưa được phân loại

- Kết quả của hàm này là 1 label được gán cho document đó, tương ứng với category mà bạn quy định từ trước.
- Để tìm hiểu kỹ hơn về KNN trong sklearn các bạn có thể tham khảo link sau:

  http://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html

  http://scikit-learn.org/stable/modules/neighbors.html#classification