Python Unstructured Learning - Building a Book Recommender Engine

Alex Knorr

```
import numpy as np
import pandas as pd
from sklearn.decomposition import NMF
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.preprocessing import normalize

from gutenberg.acquire import load_etext
from gutenberg.cleanup import strip_headers
from gutenberg.query import get_etexts
from gutenberg.query import get_metadata

#Load books from Gutenberg project#
MD = strip_headers(load_etext(2701).strip())
VK = strip_headers(load_etext(2400).strip())
ARG = strip_headers(load_etext(2703).strip())
LO = strip_headers(load_etext(2730).strip())
DT = strip_headers(load_etext(1001).strip())
HEN = strip_headers(load_etext(1100).strip())
DO = strip_headers(load_etext(501).strip())
SH = strip_headers(load_etext(41).strip())
MET = strip_headers(load_etext(5421).strip())
BER = strip_headers(load_etext(12016).strip())

#Create a list of the text
books = [MD, VK, ARG, LO, DT, HEN, DO, SH, MET, BER]

#Create a list of the book titles
title = ['Moby Dick','Vikram and the Vampire', 'The Argonauts of North Liberty', 'Long Odds', 'The Divin
         'The First Part of Henry the Sixth', 'The Story of Doctor Dolittle',
         'The Legend of Sleepy Hollow', 'The Metropolis', 'The Merchant of Berlin']

#Convert to CSR matrix (needs to be list)#
tfidf = TfidfVectorizer()
csr_mat = tfidf.fit_transform(books)
print(csr_mat.toarray())

words = tfidf.get_feature_names()
print(words)

#Create NMF features#
model = NMF(n_components=4)
#arbitray number of components, but based on 4 books in 2000s, 2 books in 1000s, 2 books less than 1000
model.fit(csr_mat)
nmf_features = model.transform(csr_mat)
print(nmf_features)
```

```python
#Convert features set and look for cosine similarities#
norm_features = normalize(nmf_features)
books_df = pd.DataFrame(norm_features, index = title)
article = books_df.loc['Moby Dick']
similarities = books_df.dot(article)

print(similarities.nlargest())
```

```
## [[ 0.00220866  0.          0.00038646 ...,  0.00012991  0.          0.        ]
##  [ 0.00140888  0.00066293  0.0012326  ...,  0.          0.          0.        ]
##  [ 0.          0.          0.         ...,  0.          0.          0.        ]
##  ...,
##  [ 0.          0.          0.         ...,  0.          0.          0.        ]
##  [ 0.          0.          0.         ...,  0.          0.          0.        ]
##  [ 0.          0.          0.00016869 ...,  0.          0.          0.        ]]
## ['000', '0001', '10', '100', '101', '102', '103', '104', '105', '106', '107', '108', '109', '11', '1
```

es', 'wards', 'ware', 'warehouse', 'warehouses', 'warensboro', 'warfare', 'warlike', 'warm', 'warmed