

Lifebloom

About R, Python, SAS, Machine Learning, Data Mining and miscellaneous things

[홈](#) [Profile](#) [Contact](#) [R](#) [Python](#) [Visualization](#) [misc](#)

Python을 활용한 텍스트 마이닝 5. 텍스트 분석-데이터 분석

시각화에서 데이터가 어떤 특징을 지니고 있는지 확인했으니 이번에는 영화 리뷰 간 유사성을 계산해보겠습니다. 이와 같은 분석 말고도 영화 평에 대한 감성 분석과 같은 여러 분석 방법들이 있습니다.

먼저 분석하려는 영화의 리뷰들을 String 형식의 데이터로 받아보겠습니다.

```
review_list=[]
for n in range(30):
    url = http://movie.daum.net/moviedb/grade?
movieid=97728&type=netizen&page=
    {}.format(n+1)
    webpage = urlopen(url)
    source =
    BeautifulSoup(webpage, 'html.parser', from_encoding='
    utf-8')
    reviews = source.findAll('p', {'class': 'desc_review'})
    for review in reviews:
        review_list.append(review.get_text().strip().replace('\n',
        '').replace('\t', '').replace('\r', ''))

    file = open('okja.txt', 'w', encoding='utf-8')

    for review in review_list:
        file.write(review + '\n')

    file.close()

doc1 = "
File failed to load: /extensions/MathMenu.js
times = me.requires()
```

Search



Recently posted

[Recommendation System 1.](#) 5월 13, 2018

[Regression with Machine Learning 4. Regularization for sparsity\(희소 학습\)](#) 4월 8, 2018

[경사 하강법\(Gradient Descent\)](#) 3월 27, 2018

[Regression with Machine Learning 3. Constrained Least Squares\(제약 최소제곱\)](#) 3월 18, 2018

[Regression with Machine Learning 2. Stochastic Gradient Descent\(확률적 경사법\)](#) 3월 18, 2018

Posts

[2018년 5월](#) (1)

[2018년 4월](#) (1)



for line in lines:

doc1 += line

file.close()

영화 '옥자'의 리뷰를 받아와서 doc1에 저장하였습니다. 이처럼 나머지 영화들도 doc2, doc3 과 같은 형식으로 저장했습니다.

```
from sklearn.feature_extraction.text import
TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity
corpus = [doc1, doc2, doc3]
vectorizer = TfidfVectorizer()
X = vectorizer.fit_transform(corpus)
X=X.todense()
```

sklearn 모듈을 이용해서 TF-IDF로 벡터화를 하고 코사인 유사도를 사용합니다. 세 리뷰의 명사들 간의 코사인 유사도를 활용해서 유사성이 얼마나 있는지 알아보겠습니다.

```
print("similarity between 'okja' and
'monster':",cosine_similarity(X[0],X[1]))
print("similarity between 'okja' and
'real':",cosine_similarity(X[0],X[2]))
print("similarity between 'monster' and
'real':",cosine_similarity(X[1],X[2]))

>similarity between 'okja' and 'monster': [[
0.26134023]]

>similarity between 'okja' and 'real': [[ 0.3445885]]

>similarity between 'monster' and 'real': [[
0.27664574]]
```

File failed to load: /extensions/MathMenu.js

2018년 3월 (5)

2018년 1월 (1)

2017년 9월 (2)

2017년 8월 (6)

2017년 7월 (12)

2017년 6월 (7)

Etc

사이트 관리

로그아웃

글 RSS

댓글 RSS

WordPress.org

category

misc (2)

Python (17)

coding with python (1)

installation (5)

Neural Network (1)

Text Mining (10)

R (14)

2017 Weather Contest (4)

machine learning (5)

Packages & Base (2)

Recommendation System (2)

Visualization (2)



Manage

구한 코사인 유사도를 프린트하면 다음과 같은 결과가 나오게 됩니다.

👤 kis0403 📅 7월 5, 2017 📁 Text Mining 💬 댓글 없음
✎ 편집

← Python을 활용한 텍스트 마이닝 4. 텍스트 분석-데이터 시각화

2017 날씨 빅데이터 콘테스트 2. 데이터 탐색(EDA) →

답글 남기기

[kis0403로\(으로\) 로그인 함.](#) [로그아웃?](#)

댓글

댓글 달기