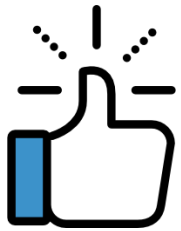
# Python
# 도서 분야 예측 시스템

아이티윌 빅데이터31기 김동환

# 목 차

CONTENTS

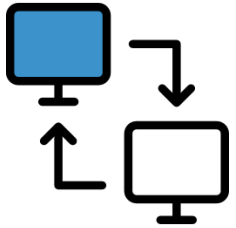


배경 및 목표

CONTENTS

구현 순서

CONTENTS



구현 내용

CONTENTS

결론

# 배 경




# 목 표

각 분야별 도서의 제목과 소개 내용을 추출 및 학습하고 분석하여 새로운 도서의 소개 내용과 제목을 통해서 도서 분야를 예측

최종 목표는 소개글, 목차, 분야를 통해 현재 상황에 맞는 책을 추천해주는 시스템

# 구현 순서

# 수집한 데이터

| | field | text |
|---|---|---|
| 1 | novel | 불편한 편의점(15만부 기념 윈터 에디션)『불편한 편의점』 |
| 2 | novel | 어서 오세요, 휴남동 서점입니다 "책과 서점을 통해 새로운 |
| 3 | novel | 디 에센셜 헤르만 헤세(교보문고 특별판) ■ 당신이 지금 만 |
| 4 | novel | 달러구트 꿈 백화점(100만부 기념 합본호: Gift Edition) 한국 |
| 5 | novel | 센 강의 이름 모를 여인《센 강의 이름 모를 여인》은 한국어 |
| 6 | novel | 미드나잇 라이브러리 ★★★★★국내 주요 서점 종합 베스. |
| 7 | novel | 밝은 밤 공감을 불러일으키는 이야기와 서정적이며 사려 깊 |
| 8 | novel | 달러구트 꿈 백화점. 2 100만 독자를 사로잡은《달러구트 꿈 |
| 9 | novel | 인간 실격 오직 순수함만을 갈망하던 여린 심성의 한 젊은이 |
| 10 | novel | 아몬드 영화와도 같은 강렬한 사건과 매혹적인 문체로 시선 |
| 11 | novel | 지구 끝의 온실 이미 폭넓은 독자층을 형성하며 열렬한 사랑 |
| 12 | novel | 백광 독자와 평단은 물론 동료 작가들로부터 명실공히 천재 |

book 데이터프레임

각 도서 분야별 150개
베스트셀러 제목, 소개글

'art_popularculture', 'computer_it',
'economy',
'history_culture','humanities',
'novel', 'poem_essay',
'politics_society', 'religion',
'selfdevelope', 'sience',
'technology_engineering'

| | | |
|---|---|---|
| 1790 | computer_ | Android Studio를 활용한 안드로이드 프로그래밍 실습 |
| 1791 | computer_ | 나의 첫 블렌더《나의 첫 블렌더》는 모델링부터 애니메 |
| 1792 | computer_ | 클린 아키텍처 우리 모두는 낮은 개발 비용으로 유연하 |
| 1793 | computer_ | 머신러닝 디자인 패턴 디자인 패턴이란 전문가 수백 명 |
| 1794 | computer_ | 레트로의 유니티 게임 프로그래밍 에센스 이 책은 기본 |
| 1795 | computer_ | IT 엔지니어를 위한 네트워크 입문 클라우드/데브옵스 |
| 1796 | computer_ | 실전 카프카 개발부터 운영까지 국내 최초이자 유일한 |
| 1797 | computer_ | 이기적 컴퓨터그래픽스운용기능사 실기 세트 본 도서는 |
| 1798 | computer_ | 시스코 네트워킹 2002년 출간 이후 16년 동안 네트워크 |
| 1799 | computer_ | C++ Programming C++는 1979년 Bjarne Stroustrup |
| 1800 | computer_ | 자료구조 ▶ 이 책은 자료구조를 다룬 이론서입니다. |

# 학습, 테스트 데이터 분류

```
x_train,x_test,y_train,y_test = train_test_split(book['text'],book['field'],test_size=0.2)
```

학습데이터

테스트데이터

```
In [10]: Counter(y_train)
Out[10]:
Counter({'humanities': 118,
         'art_popularculture': 122,
         'religion': 124,
         'computer_it': 121,
         'politics_society': 120,
         'poem_essay': 126,
         'history_culture': 115,
         'economy': 118,
         'technology_engineering':
123,
         'novel': 115,
         'selfdevelope': 118,
         'sience': 120})
```

```
In [11]: Counter(y_test)
Out[11]:
Counter({'poem_essay': 24,
         'history_culture': 35,
         'humanities': 32,
         'sience': 30,
         'religion': 26,
         'economy': 32,
         'novel': 35,
         'selfdevelope': 32,
         'politics_society': 30,
         'art_popularculture': 28,
         'computer_it': 29,
         'technology_engineering':
27})
```

# 전처리, 형태소 분석

From Konlpy.tag
Import Okt

형태소 분석 라이브러리
Okt 활용

명사 : Noun
형용사 : Adjective
영어 : Alpha

사용된 형태소

학습에 도움되지 않는
불용어

['지난','이후','독자','작가','있다','같은','이로','인해','있게','누구','있으며','있는','통해','있도록','저자']

# 학습

```
cv = CountVectorizer(tokenizer=okt_pos,stop_words)
```

CountVectorizer를 사용하여 학습하여 cv변수에 저장

```
def okt_pos(arg):
    token = []
    for j in okt.pos(arg):
        if j[1] in ['Noun','Adjective','Alpha']:
            token.append(j[0])
    token = [i for i in token if len(i)>=2]
    return token
```

토큰화 적용할 함수 : okt_pos

- 2글자 이상의 명사, 형용사, 영어만 추출

# 테스트 데이터 적용 및 예측

```
x_test = cv.transform(x_test)
```

테스트 데이터 적용

```
nb = MultinomialNB()
nb.fit(x_train,y_train)
```

분류기 모델에
학습데이터로 모델화

```
y_predict = nb.predict(x_test)
sum(y_predict == y_test)/360 # 정답률
accuracy_score(y_test,y_predict)
```

예측

```
360 # 정답률
Out[28]: 0.7888888888888889
```

정답률

# 혼동 행렬

```
pd.crosstab(y_test,y_predict)
confusion_matrix(y_test,y_predict)
```

```
In [33]: confusion_matrix(y_test,y_predict)
Out[33]:
array([[24,  0,  1,  0,  1,  0,  0,  0,  0,  1,  0,  1],
       [ 0, 26,  0,  0,  0,  0,  0,  0,  0,  1,  0,  2],
       [ 0,  1, 27,  0,  2,  0,  0,  1,  0,  1,  0,  0],
       [ 0,  0,  0, 29,  1,  1,  0,  2,  0,  2,  0,  0],
       [ 0,  3,  1,  2, 18,  0,  1,  2,  0,  4,  1,  0],
       [ 0,  0,  0,  1,  1, 28,  5,  0,  0,  0,  0,  0],
       [ 0,  0,  0,  0,  2,  1, 17,  0,  0,  3,  1,  0],
       [ 0,  0,  2,  1,  1,  0,  1, 20,  0,  2,  0,  3],
       [ 0,  0,  0,  0,  0,  1,  0,  1, 23,  1,  0,  0],
       [ 0,  2,  0,  0,  0,  0,  1,  1,  0, 28,  0,  0],
       [ 0,  0,  0,  0, 10,  1,  0,  0,  0,  0, 19,  0],
       [ 0,  0,  1,  0,  0,  0,  0,  0,  0,  0,  1, 25]], dtype=int64)
```

Confusion_matrix

| col_0 | art_popularculture | computer_it | economy | history_culture | humanities | novel | poem_essay | politics_society | religion | selfdevelope | sience | technology_engineering |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| art_popularculture | 24 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| computer_it | 0 | 26 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 2 |
| economy | 0 | 1 | 27 | 0 | 2 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| history_culture | 0 | 0 | 0 | 29 | 1 | 1 | 0 | 2 | 0 | 2 | 0 | 0 |
| humanities | 0 | 3 | 1 | 2 | 18 | 0 | 1 | 2 | 0 | 4 | 1 | 0 |
| novel | 0 | 0 | 0 | 1 | 1 | 28 | 5 | 0 | 0 | 0 | 0 | 0 |
| poem_essay | 0 | 0 | 0 | 0 | 2 | 1 | 17 | 0 | 0 | 3 | 1 | 0 |
| politics_society | 0 | 0 | 2 | 1 | 1 | 0 | 1 | 20 | 0 | 2 | 0 | 3 |
| religion | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 23 | 1 | 0 | 0 |
| selfdevelope | 0 | 2 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 28 | 0 | 0 |
| sience | 0 | 0 | 0 | 0 | 10 | 1 | 0 | 0 | 0 | 0 | 19 | 0 |
| technology_engineering | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 25 |

crosstab

# 결과 데이터

print(classification_report(y_test,y_predict))

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| art_popularculture | 1.00 | 0.86 | 0.92 | 28 |
| computer_it | 0.81 | 0.90 | 0.85 | 29 |
| economy | 0.84 | 0.84 | 0.84 | 32 |
| history_culture | 0.88 | 0.83 | 0.85 | 35 |
| humanities | 0.50 | 0.56 | 0.53 | 32 |
| novel | 0.88 | 0.80 | 0.84 | 35 |
| poem_essay | 0.68 | 0.71 | 0.69 | 24 |
| politics_society | 0.74 | 0.67 | 0.70 | 30 |
| religion | 1.00 | 0.88 | 0.94 | 26 |
| selfdevelope | 0.65 | 0.88 | 0.75 | 32 |
| sience | 0.86 | 0.63 | 0.73 | 30 |
| technology_engineering | 0.81 | 0.93 | 0.86 | 27 |
| accuracy | | | 0.79 | 360 |
| macro avg | 0.80 | 0.79 | 0.79 | 360 |
| weighted avg | 0.80 | 0.79 | 0.79 | 360 |

Accuracy : 정확도
Precision : 정밀도
Recall : 재현율
F1-score : 정밀도+재현율

# 시각화

computer_it


sience


religion


novel


technology_engineering


art_popularculture

# 시각화

politics_society


history_culture


economy


humanities


poem_essay


selfdevelope

# 감사합니다