2040년 07월 10일 미리캔버스

[5팀 파워레인조]

뉴스 제목에 따른 토픽 분류 모델 개발 및 적용

이시현 정승욱 최보근 김윤민 이민형

목차

오늘의 정보안내

1 분석 주제 & 목표

2 서론: 데이터 설명 & EDA

3 Phase 01

4 Phase 02

- 재 전처리 후 모델링. . 최종 모델 선택

5 서비스 Stream lit

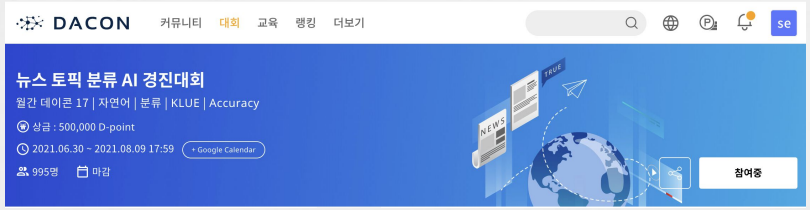
[주제 및 목표] 뉴스 토픽 분류 모델 개발 및 적용

[목표]

1. 데이콘 데이터 셋 활용

2. 뉴스 제목으로 태깅된 토픽 학습 후 토픽 추정

3. 서비스화 : 네이버 뉴스 기사 헤드라인 또는 제목 스크래핑 후 적용



[타임라인]

✅ 화: 데이터, 주제 정하기 + EDA

✅ 수: EDA, 전처리, 간단한 모델링

✅ 목: 모델링 심화, 시각화 / 중간발표

✅ 금: 불용어 전처리, LDA, 서비스 개발

✅ 월: ppt, 시각화, 서비스, 자료 합치기

[서론] 데이터 설명 : YNAT 데이터 세트

YNAT:

주제분류를 위한 연합뉴스 헤드라인

Supervised ML NLP, KLUE Data!

목적:

데이터 세트로 학습시킨 모델을 만들어

원하는 기사 제목의 토픽(주제)을 추정해보기

```
topic_dict
```

	topic	topic_idx
0	IT과학	0
1	경제	1
2	사회	2
3	생활문화	3
4	세계	4
5	스포츠	5
6	정치	6

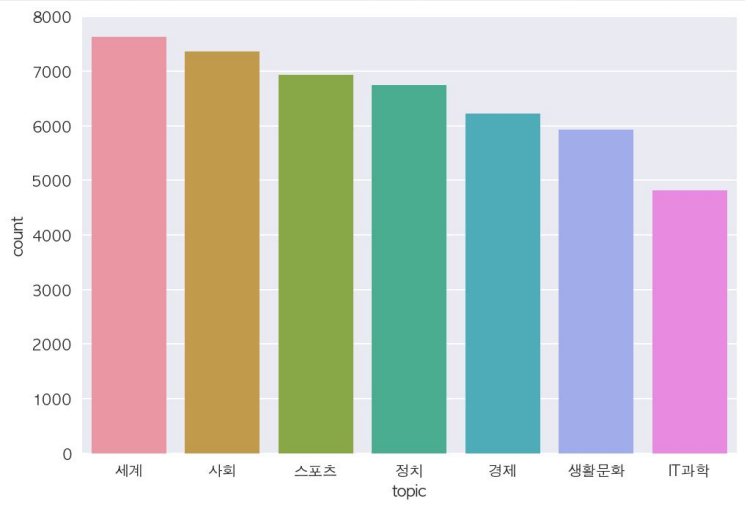
```
train.head(5).set_index('index')
```

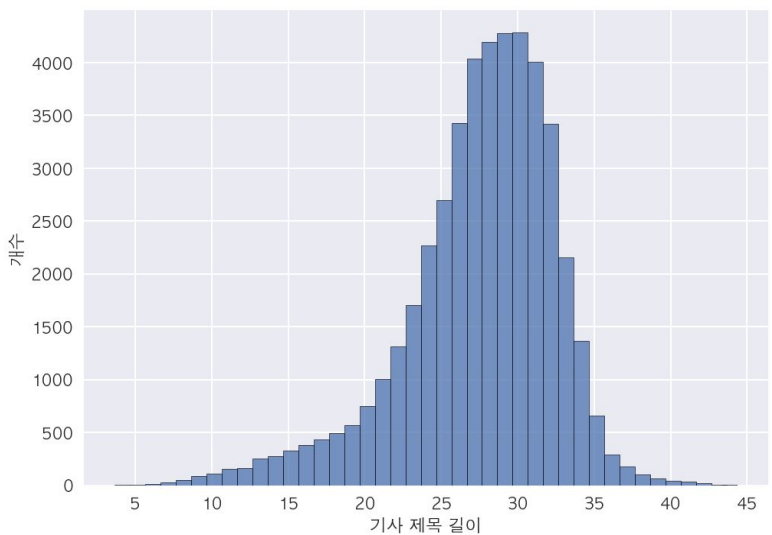
index	title	topic_idx
0	인천→핀란드 항공기 결항…휴가철 여행객 분통	4
1	실리콘밸리 넘어서겠다…구글 15조원 들여 美전역 거점화	4
2	이란 외무 긴장완화 해결책은 미국이 경제전쟁 멈추는 것	4
3	NYT 클린턴 측근韓기업 특수관계 조명…공과 사 맞물려종합	4
4	시진핑 트럼프에 중미 무역협상 조속 타결 희망	4

```
test.head(3).set_index('index')
```

index	title
45654	유튜브 내달 2일까지 크리에이터 지원 공간 운영
45655	어버이날 맑다가 흐려져…남부지방 옅은 황사
45656	내년부터 국가RD 평가 때 논문건수는 반영 않는다

[서론] 간단한 EDA - 토픽 별 기사 건수 & 제목 길이





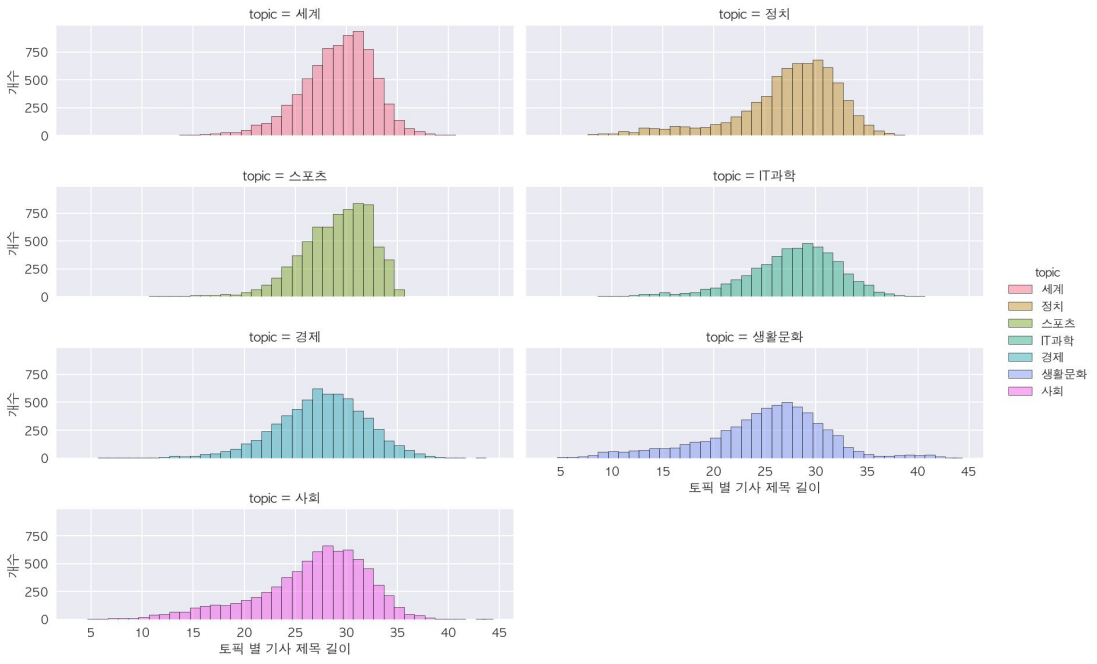
모델 학습 train, test set 분리 시

계층 추출 필요 (Stratified Random)

기사 제목 길이 분포: 종 모양 형태를 띄며 일정

(Mean 27.3 IQR 6 Std 1.9 Max 44)

[서론] 간단한 EDA - 토픽 별 문장 길이



차이 크지 않고 일정

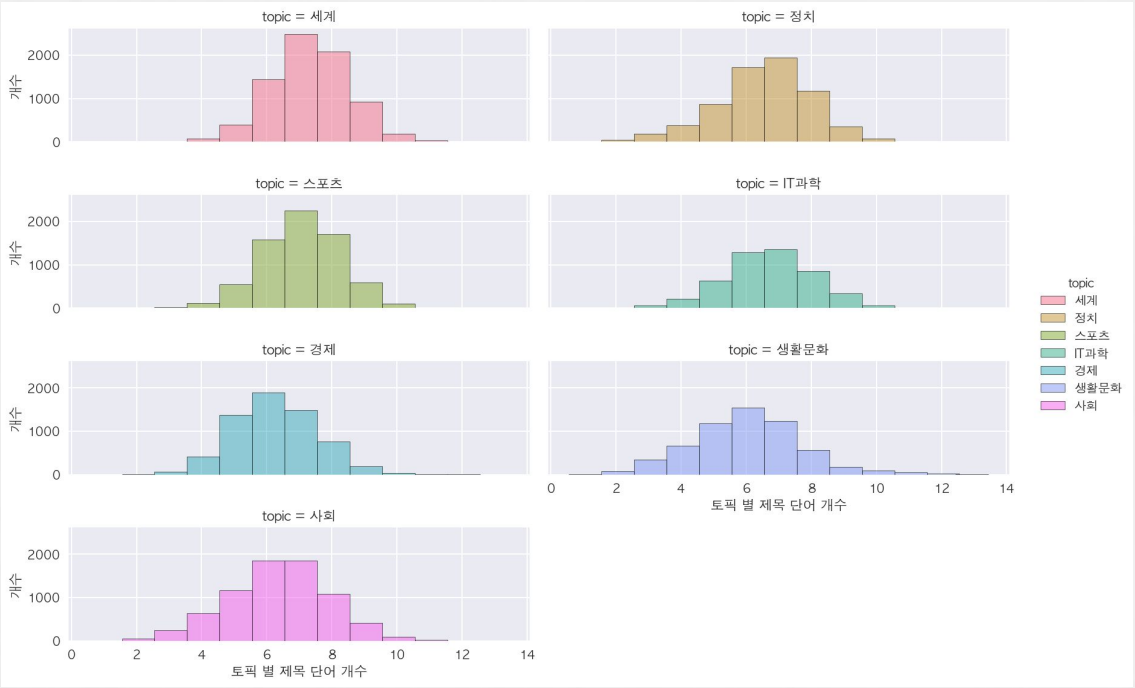
토픽 간 비교를 위해

스케일링은

필요하지 않다고 판단

(로그변환, 표준화 등)

[서론] 간단한 EDA - 토픽 별 단어 개수



차이 크지 않고 일정

토픽 간 비교를 위해

스케일링은

필요하지 않다고 판단

(로그변환, 표준화 등)

2040년 07월 10일

Phase 01

잘 나올 거야 잘 나오겠지

[Phase 01] 데이터 전처리 - 한자 대체

```python
# Check Chinese characters with high frequency of appearance
# from collections import Counter
k = []
for i in range(0, len(train)):
    a = re.findall('[一-龥]', train['title'][i])
    if len(a) != 0:
        k = [*k, *a]
Counter(k).most_common()[:10]
```

```
[('美', 1498),
 ('北', 1329),
 ('中', 795),
 ('朴', 661),
 ('日', 467),
 ('青', 381),
 ('與', 291),
 ('英', 285),
 ('文', 184),
 ('野', 181)]
```

뉴스 제목: 한자 의미 중요 & **多**

상위 30개 한자 해석 사용

```python
# 일부 한자 수정
name = {'↑':"상승",'↓':"하락",'㈜':"","銀":"은행","外人":"외국인",
        "日":"일본","美":"미국","北":"북한","英":"영국","中":"중국",
        "伊":"이탈리아","韓":"한국","南":"한국","獨":"독일","佛":"프랑스",
        "亞":"아시아", "與":"여당","靑":"청와대","野":"야당","檢":"검찰",
        "銀":"은행",,"人":"사람","企":"기업","前":"이전","車":"자동차",
        "軍":"군대","朴":"박근혜", "文":"문재인", "安":"안철수","展":"전시회",
        "反":"반대", "故":"사망", "男":"남자", "女" : "여자",
        "硏":"연구", "코로나 19":"코로나19","19":"코로나"}
for i, j in name.items():
    text = text.replace(i, j)
```

initial code

[Phase 01] 데이터 전처리

✅ 토큰화 : Okt

✅ 품사 태깅 : 명사, 알파벳, 형용사, 동사만 사용

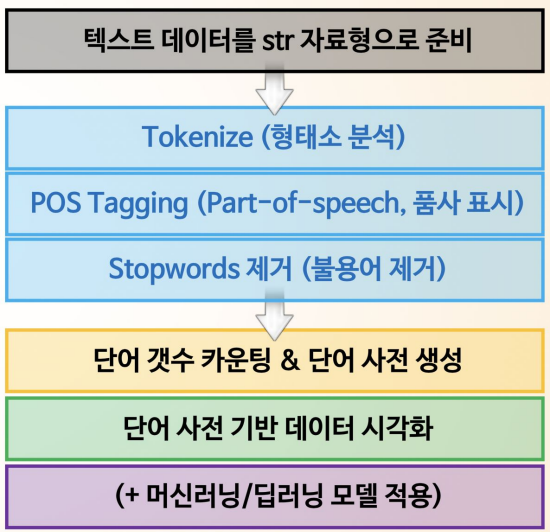
✅ 텍스트 전처리

- 개행문자 제거
- 한자 대체
- 한글, 영문만 남김
- 중복으로 생성된 공백값 제거
- 영문자 소문자

✅ 불용어 제거

- 많이 나오지만 분석에 도움 되지않는 단어 제거
- 한 글자 단어 : 제외

[Phase 01] 단어 사전 생성 - tf-idf matrix




```
# subliner_tf =True: tf scaling 1 + log(tf) 스케일링: 높아짐
# norm: {'l1', 'l2'}, default = 'l2', 'l1'norm은 오히려 떨어짐
# strip_accents='unicode': 영향 없음
# min_df: DF(document-frequency: 문서의 수)의 최소 빈도값 설정 : 낮아짐
# analyzer = 'word' 'char' : 'word': 영향 없

tfidf_vect = TfidfVectorizer(tokenizer=split, sublinear_tf=True, norm = 'l2', analyzer = 'word')
tfidf_vect.fit(train2.corpus)

tfidf_matrix_train = tfidf_vect.transform(train2.corpus)

print(tfidf_matrix_train.shape)

#tfidf_matrix_test = tfidf_vect.transform(test['title'])
#print(tfidf_matrix_test.shape)
```

```
(45654, 26036)
```

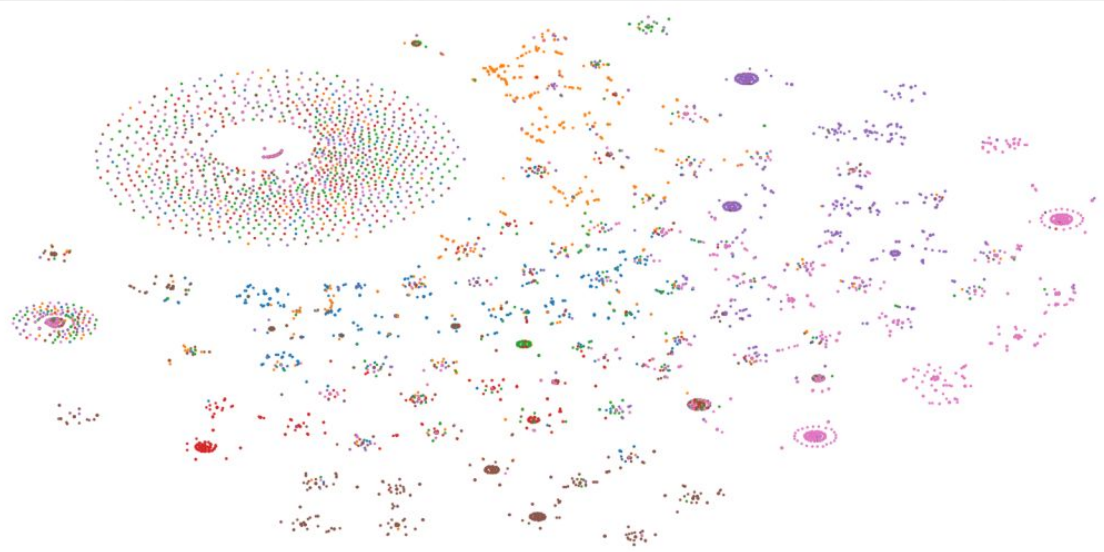
✅ tf-idf 선정 이유: 특정 단어가 토픽을 드러냄

✅ analyzer = ‘word’ : 단어 단위로 분석 적용

✅ sublinear_tf = True: 단어의 빈도수 1+log(tf)로 스케일링

✅ Count-vec.와 성능비교시 tf-idf가 성능이 더 좋음

[Phase 01] 단어 사전 기반 데이터 시각화 - t-SNE



✅ test data의 토픽 예측 결과 t-SNE로 시각화

[시각화 결과]

✅ 오른쪽 부분 잘 분류함

✅ 왼쪽 상단 분류하지 못함

- 동그랗게 여러 색의 점 모여있는 것 확인

[결론]

다시 데이터 전처리 필요!!

[Phase 01] LDA 차원축소

```
display_topics(lda_tfidf, feature_names_tfidf, 15)
```

```
Topic # 0
경기 감독 코스피 월드컵 시즌 류현진 mlb 축구 게임 홈런 아시안 손흥민 종합 리그 연속
Topic # 1
미국 종합 프로농구 사망 현대 nba 이란 여자배구 시위 테러 북한 감독 꺾다 연승 한국
Topic # 2
종합 미국 북한 대통령 이란 민주 합의 장관 트럼프 국회 정부 제재 청와대 회의 터키
Topic # 3
게시판 코로나 개발 투자 ai 금융 기술 네이버 지원 한국 기업 종합 개최 그래픽 사업
Topic # 4
대통령 북한 박근혜 종합 중국 정상회담 평양 남북 미국 코로나 김정은 정상 한국 청와대 트럼프
Topic # 5
날씨 축제 여행 주말 전국 주의보 서울 종합 신간 강원 오후 최고 충북 내일 기온
Topic # 6
억원 분기 영업 출시 삼성 lg kt 전자 작년 sk 종합 증권 상승 증가 이익
```

LDA:

분포를 가정하고 잠재적인 의미(토픽)들을 찾음

개별 클래스를 분별할 수 있는 기준을

최대한 유지하며 차원 축소

[결론] 다시 데이터 전처리 필요

```
import sys
import numpy as np

# 동적 변수 생성 코드

mod = sys.modules[__name__]
for topic in topic_list:
    setattr(mod, 'df_{}'.format(topic), df[df.topic == topic])
    setattr(mod, 'corpus_{}'.format(topic),
            ' '.join(getattr(mod,'df_{}'.format(topic)).corpus.tolist()).split(' '))
    setattr(mod, 'most_{}'.format(topic),
            dict(Counter(getattr(mod,'corpus_{}'.format(topic))).most_common(100)))
    setattr(mod, 'least_{}'.format(topic),
            dict(list(dict(Counter(getattr(mod,'corpus_{}'.format(topic))).most_common()).items())[-100:]))
```

```
IT과학 뉴스와 경제 뉴스의 공통 단어 : 25개
IT과학 뉴스와 사회 뉴스의 공통 단어 : 13개
IT과학 뉴스와 생활문화 뉴스의 공통 단어 : 12개
IT과학 뉴스와 세계 뉴스의 공통 단어 : 9개
IT과학 뉴스와 스포츠 뉴스의 공통 단어 : 10개
IT과학 뉴스와 정치 뉴스의 공통 단어 : 8개
경제 뉴스와 사회 뉴스의 공통 단어 : 17개
경제 뉴스와 생활문화 뉴스의 공통 단어 : 8개
경제 뉴스와 세계 뉴스의 공통 단어 : 10개
경제 뉴스와 스포츠 뉴스의 공통 단어 : 12개
경제 뉴스와 정치 뉴스의 공통 단어 : 8개
사회 뉴스와 생활문화 뉴스의 공통 단어 : 14개
사회 뉴스와 세계 뉴스의 공통 단어 : 14개
사회 뉴스와 스포츠 뉴스의 공통 단어 : 5개
사회 뉴스와 정치 뉴스의 공통 단어 : 16개
생활문화 뉴스와 세계 뉴스의 공통 단어 : 7개
생활문화 뉴스와 스포츠 뉴스의 공통 단어 : 8개
생활문화 뉴스와 정치 뉴스의 공통 단어 : 8개
세계 뉴스와 스포츠 뉴스의 공통 단어 : 6개
세계 뉴스와 정치 뉴스의 공통 단어 : 30개
스포츠 뉴스와 정치 뉴스의 공통 단어 : 4개
```

[Phase 01] 모델 성능 비교 - 단어 벡터 + 모델

모델	성능
TF-IDF + LGBM	0.8109
TF-IDF + Logistic Reg	0.8449
TF-IDF + Naive Bayes	0.8368
TF-IDF + Linear SVC	0.8476
Count Vec + LGBM	0.8185
Count Vec + Logistic Reg	0.8401

[희소행렬 다중 분류를

효과적으로 처리하는 알고리즘]

✅ 로지스틱 회귀 (Logistic Regression)

✅ 서포트 벡터머신 (SVC)

- Kernel : **Linear**, RBF (가우시안)

✅ 나이브 베이즈 (Naive Bayes)

다양한 분류모델 적용 결과

LinearSVC의 성능이 가장 높음

[Phase 01] 모델링

LinearSVC모델에 StratifiedKfold 적용시 0.8476 > 0.8484까지 정확도 증가

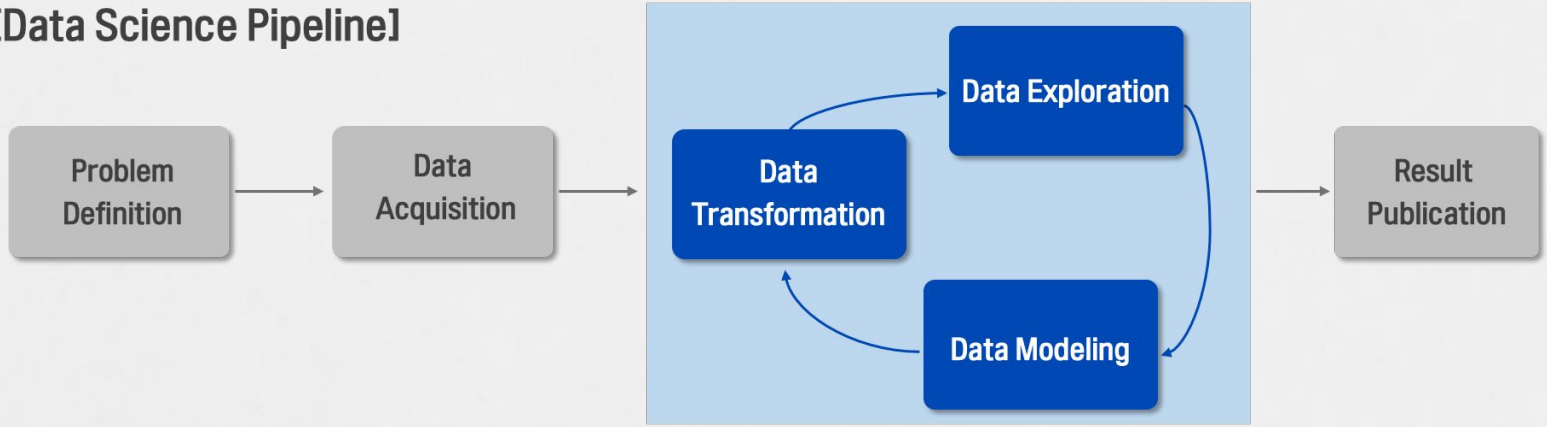
하지만 이 이상으로 성능 증가 하지 않음

```
# LinearSVC K-fold
s_kf = StratifiedKFold(n_splits = 10, shuffle = True, random_state = 0)

best_accuracy = 0

accuracy_ = []
for train_index in s_kf.split(X_train,y_train):

    model = LinearSVC( C=0.1,tol=0.1, max_iter=50, verbose = 2, random_state=0)
    model.fit(X_train, y_train)

    y_pred = model.predict(X_test)
    accuracy_.append(accuracy_score(y_pred, y_test))
print("₩n max 정확도 :", np.max(accuracy_))
```

```
[LibLinear][LibLinear][LibLinear][LibLinear][LibLinear][LibLinear][LibLinear][LibLinear][LibLinear][LibLinear]
 max 정확도 : 0.8484339636416733
```

[성능 향상 벽에 부딪히다] 재검토, 다시 돌아가자

[Data Science Pipeline]



모델 하이퍼파라미터 조정해도 정확도 향상 ❌

+ 강사님, 클래스매니저님, 서포터님 피드백
+ LDA, t-SNE 시각화

불용어 처리 더 하자

2040년 07월 10일

Phase 02

끝없는 불용어 처리 😭

데이터전처리_최종.ipynb

[Phase 02] 데이터 전처리 - 불용어 처리

[기존 데이터셋]

	index	title	topic_idx	corpus
0	0	인천→핀란드 항공기 결항…휴가철 여행객 분통	4	인천 핀란드 항공기 결항 휴가 여행객 분통
1	1	실리콘밸리 넘어서겠다…구글 15조원 들여 美전역 거점화	4	실리콘밸리 넘어서다 구글 조원 들이다 미국 전역 거점
2	2	이란 외무 긴장완화 해결책은 미국이 경제전쟁 멈추는 것	4	이란 외무 긴장 완화 해결 미국 경제 전쟁 멈추다
3	3	NYT 클린턴 측근韓기업 특수관계 조명…공과 사 맞물려종합	4	nyt 클린턴 측근 한국 기업 특수 관계 조명 공과 맞다 물리다 종합
4	4	시진핑 트럼프에 중미 무역협상 조속 타결 희망	4	시진핑 트럼프 중미 무역 협상 조속 타결 희망

- 자주 쓰이는 한자를 한글로 대체
- 영문자는 소문자로 통일
- 공백값 제거
- del_list를 통해 불용어 제거 & 한 글자인 단어들도 제거
- 품사 태깅을 활용하여 [명사, 형용사, 동사, 영단어]만 남기기

[변경사항 01]

- 동사 제외 :
  - 핵심 단어들이 명사가 주, 분류에 도움안되는 동사 多 ➡️ 동사 제외
- 모든 토픽에서 자주 등장하는 단어 선별해서 제외
  - 카테고리별 단어 개수 파악 (by. collections / Counter 함수)
  - 상위 300개 단어 추출 (by. most_commons 함수)
  - 공통 단어 추출 (by. 토픽별 교집합(intersection))
  - 공통 단어
    ('종합','한국','내년','없다','내달','올해','앞두다','코로나') 제거

[문제점] - 일부 corpus NaN 생성

	index	title	topic_idx	corpus
34451	34451	봄이 왔어요	3	NaN
19610	19610	눈 감은 우병우 헤쳐나가는 것도 제 몫	2	NaN
43035	43035	둑 터지고 논 패이고	3	NaN

[Phase 02] 데이터 전처리 - 불용어 처리

[변경사항 02]

- 한 글자 단어 중 의미 있는 것 수작업으로 선별
  - corpus가 Nan으로 표시되는 값들을 보니 논, 눈, 둑 등 의미 있는 단어들이 한글자 단어라서 제거되는 경우가 발생
  - 한글자 단어 중 의미있는 단어들만 추출

	index	title	topic_idx	corpus
34451	34451	봄이 왔어요	3	봄
19610	19610	눈 감은 우병우 헤쳐나가는 것도 제 몫	2	눈
43035	43035	둑 터지고 논 패이고	3	둑 논

[문제점]

성능이 낮게 나옴

[변경사항 03] - [최종]

- 동사 포함
  - 초기 모델에 비하면 성능이 약간 낮으나 필요한 단어들이 포함되어 있다고 판단하여 최종 데이터셋으로 선정

	index	title	topic_idx	corpus
34451	34451	봄이 왔어요	3	오다
19610	19610	눈 감은 우병우 헤쳐나가는 것도 제 몫	2	감다 헤치다 나가다
43035	43035	둑 터지고 논 패이고	3	터지다

	index	title	topic_idx	corpus
34451	34451	봄이 왔어요	3	봄 오다
19610	19610	눈 감은 우병우 헤쳐나가는 것도 제 몫	2	눈 감다 헤치다 나가다
43035	43035	둑 터지고 논 패이고	3	둑 터지다 논

[Phase 02] 모델링

불용어 처리 데이터에 'TF-IDF + 단일 모델' 적용 결과

모델	성능
TF-IDF + LGBM	0.8014
TF-IDF + Logistic Reg	0.8426
TF-IDF + Naive Bayes	0.8337
TF-IDF + Linear SVC	0.8390

phase 1때 성능이 가장 좋았던 LinearSVC가 0.8476 > 0.8390으로 떨어짐

[Phase 02] 모델링 (Grid Search)

GridSearch 이용 LinearSVC 최적 파라미터 값 찾기

```
# GridSearch

from sklearn.model_selection import GridSearchCV

param_grid = {
            "C":[1,0.1,0.01],
            "tol":[0.01, 0.001, 0.1,0.0001],
            "max_iter": [50,100,200]
              }

grid = GridSearchCV(svc_clf, param_grid, refit=True, verbose=2)

grid.fit(X_train, y_train)

print('The best parameters are ', grid.best_params_)
```

```
1  grid.best_params_
```

```
{'C': 0.1, 'max_iter': 50, 'tol': 0.1}
```

[Phase 02] 모델링

최적 파라미터로 모델링한 결과 0.8466 > 0.8490 으로 향상

```
# LinearSVC K-fold(GridSearch)

s_kf = StratifiedKFold(n_splits = 10, shuffle = True, random_state = 0)

accuracy_ = []
for train_index in s_kf.split(X_train,y_train):

    model = LinearSVC( C=0.1,tol=0.1, max_iter=50, verbose = 2, random_state=0)
    model.fit(X_train, y_train) # <- x_train_transformed (not x_train)

    y_pred = model.predict(X_test) # 예측 라벨
    accuracy_.append(accuracy_score(y_pred, y_test)) # 정확도 측정 및 기록

#print("각 분할의 정확도 :", accuracy_)
print("₩n max 정확도 :", np.max(accuracy_))
```

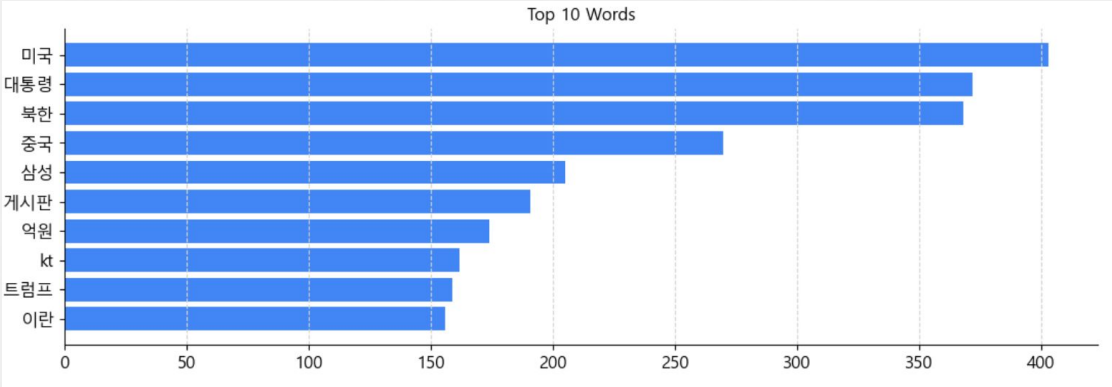
```
[LibLinear][LibLinear][LibLinear][LibLinear][LibLinear][LibLinear][LibLinear][LibLinear][LibLinear][LibLinear]
 max 정확도 : 0.846900781192962
```

2040년 07월 10일

최종 시각화

기대하십쇼 😎

상위 10개 단어 (by. Squarify)

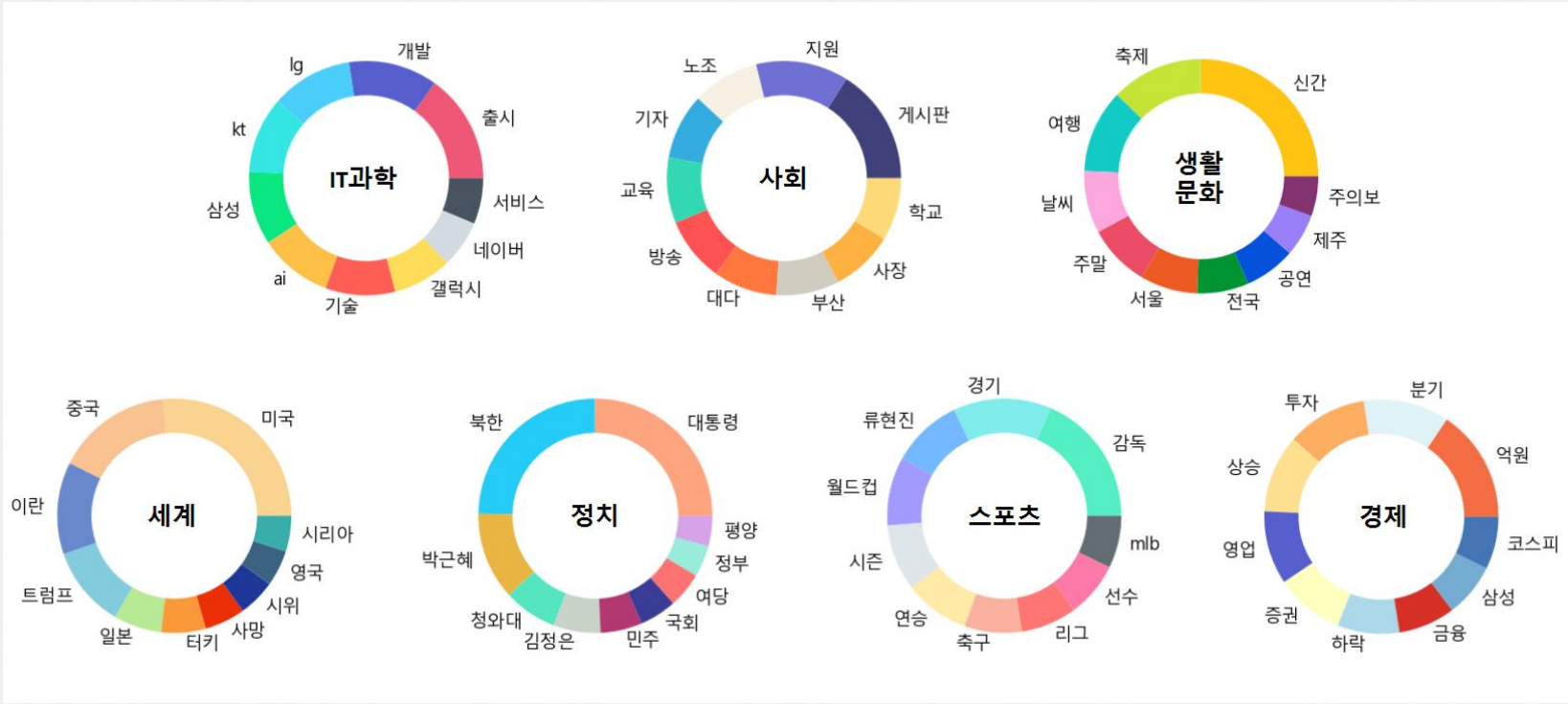


WordCloud

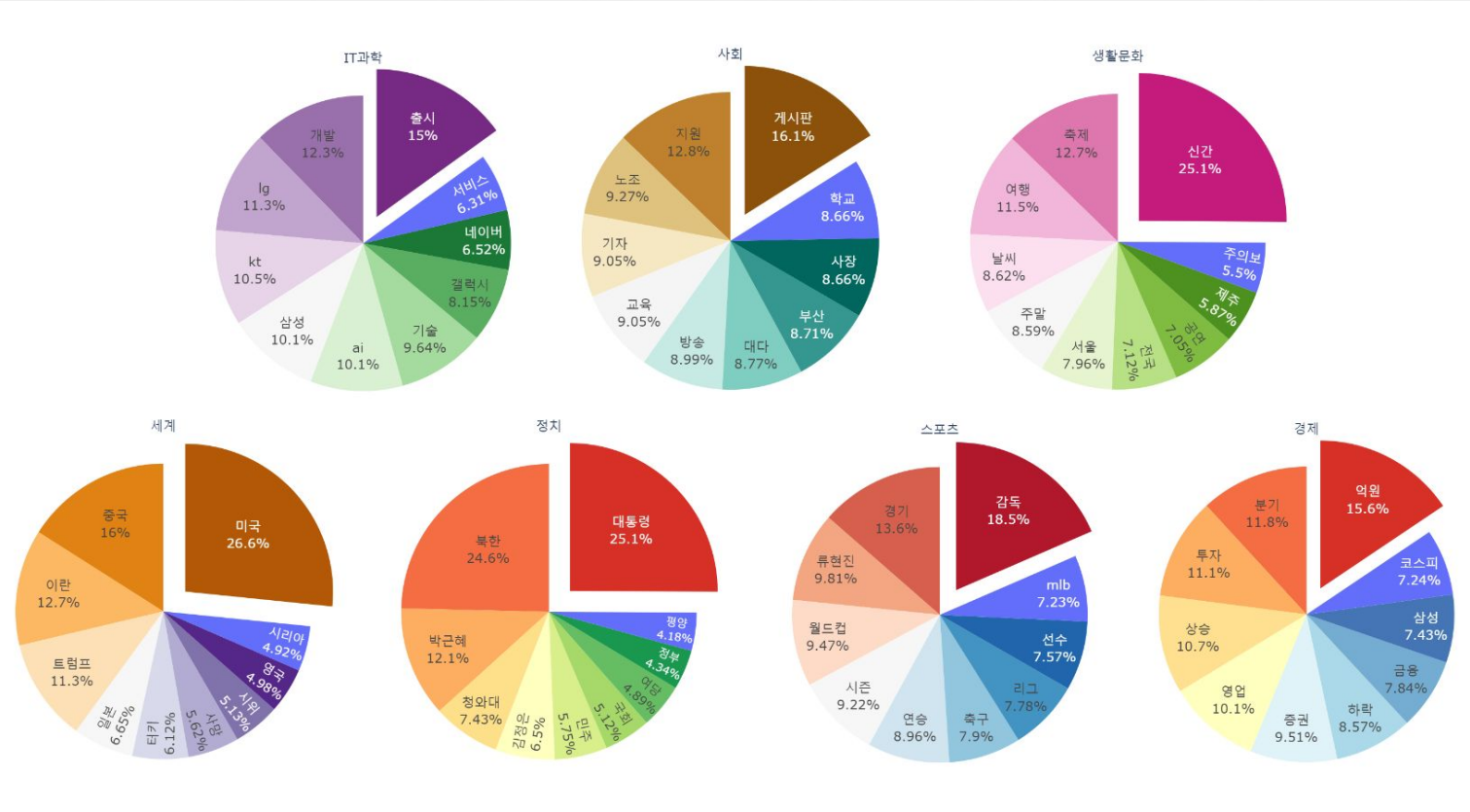

월드컵
게시판
분기
장관
일본
북한
국내
미국
투자
네이버
이란
트럼프
kt
대표
공개
경기
증권
대다
박근혜
세계
정부
서비스
최고
청와대
중국
지원
삼성
민주
감독
상승
서울
현대
시장
개최
스마트폰
작년
출시
sk
lg
축제
대회
김정은
리그
부산
게임
인터넷


출시
북한
게시판
공개
상승
개최
일본
영업
총리
중국
삼성
트럼프
네이버
작년
장관
이란
국내
하락
투자
현대
시장
감독
청와대
지원
박근혜
대다
신간
lg
민주
최고
대표
갤럭시
경기
sk
분기
kt
서울
세계
미국
금융
월드컵
증권

카테고리별 상위 10개 단어



카테고리별 상위 10개 단어들



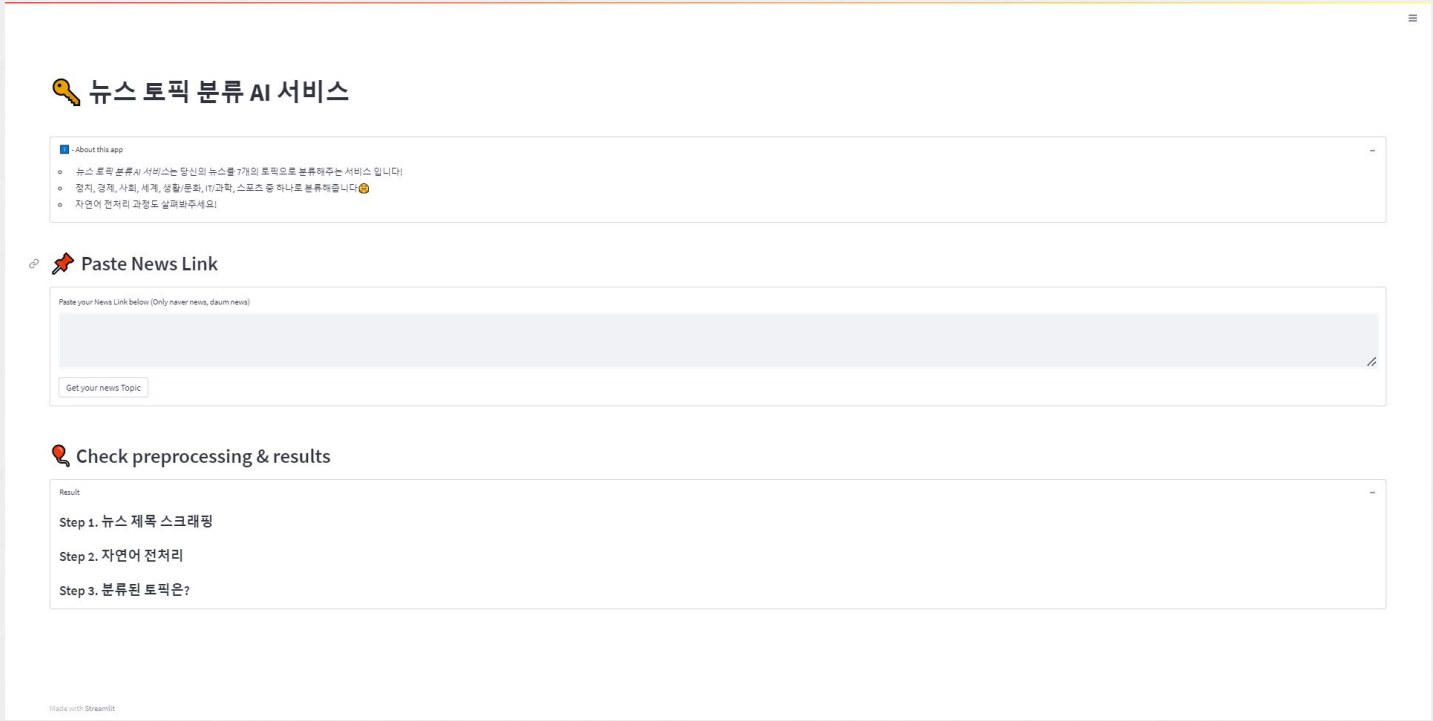
2040년 07월 10일

서비스 시연

기대하십쇼 😎

서비스 - 뉴스 URL 입력하면 토픽을 알려줘요! (Streamlit)

서비스 시연 3 - 4분



[전후 분석]

t-SNE 최종 분류 시각화

왜 잘 안됐을까?

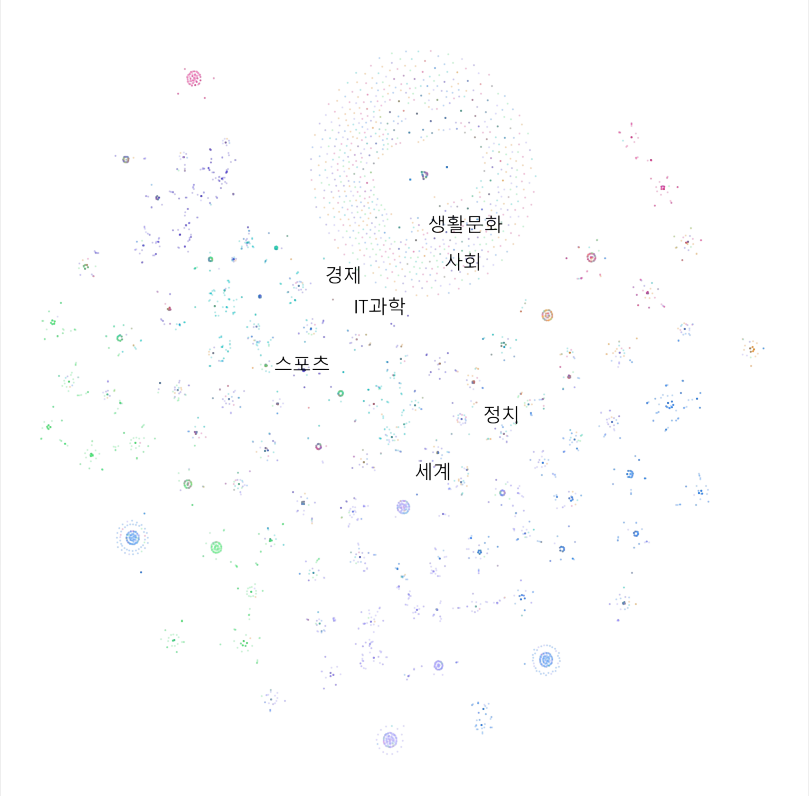
예상 원인 1.

사회, 경제 등과 같이 기준 자체에 대한 모호성이 존재.

예상 원인 2.

뉴스 헤드라인은 보편적인 단어를 사용.

공통 단어
IT과학, 경제 : 25개
세계, 정치 : 30개
생활문화, 사회 : 14개



[활용 방안]

- 기자가 뉴스 작성 후 토픽을 지정하지 않고 홈페이지에서 자동으로 분류해 게시

- 홈페이지의 토픽 분류 변경할 때마다 일일이 수작업으로 바꿔주지 않아도 됨

2040년 07월 10일

땡큐🥳