일상 대화 데이터 바탕 주제 분류

Ⅰ. 개요

> AI Hub의 '주제별 텍스트 일상 대화 데이터'를 이용해서 텍스트별 대화 주제를 예측하는 프로젝트이다. json파일 형태를 데이터프레임으로 전처리한 후 konlpy와 BOW를 이용해 텍스트 처리를 진행하였고 RandomForestClassifier을 이용해서 주제를 예측하였다.

Ⅱ. 기획

1. 분석 목표

- 텍스트 칼럼과 주제 칼럼으로 구성된 데이터프레임을 제작한 후에 주제 칼럼을 target으로 지정한다. RandomforestClassifier을 이용해서 train data를 학습시킨 후 주제를 예측한다.

2. 분석 과정

1. AI Hub에서 '**주제별 텍스트 일상 대화 데이터**'를 다운로드 한다.
2. 다운받은 데이터를 text 칼럼, subject칼럼을 가지는 데이터프레임으로 전처리한다.
3. BOW Vectorizer을 구현한다.
4. RandomforestClassifier이용해서 분류 모델링을 진행한다.

Ⅲ. 데이터

1. 데이터 목록

☑ 수집됨	Aa 데이터명	자료형	데이터 수집 방법(출처) / 메모
☐	주제별 텍스트 일상 대화 데이터	문자	데이터 분야 - AI 데이터찾기 - AI-Hub (aihub.or.kr)

2. 데이터 수집

- **주제별 텍스트 일상 대화 데이터**

소개 : 식음료, 주거와 생활, 교통, 교육, 가족 등 20여개 주제에 대한 자유로운 일상대화 텍스트를 수집한 후, 각 대화의 주제와 참여 회차 정보, 화행이 나타나는 문장에 대한 라벨링을 통해, 한국어 일상대화의 주제, 화행 등 정보를 담은 데이터셋

TL_01.KAKAO(1) ⇒ KAKAO_898_15 ⇒ 플랫폼명_주제내순서_주제번호

3. 데이터 전처리

- **원본 데이터**


```
{
  "dataset": {
    "identifier": 72966,
    "name": "KAKAO_898_15_set",
    "src_path": "/data/file/cubeManager/PROJECT001/53/txt20211005152750071856/KAKAO_898_15_set/",
    "label_path": "/data/file/cubeManager/PROJECT001/53/txt20211005152750071856/KAKAO_898_15_set/",
    "category": 2,
    "type": 0
  },
  "licenses": {
    "name": "Apache License 1.0",
    "url": "http://www.apache.org/licenses/LICENSE-1.0"
  },
  "info": [
    {
      "id": 41272,
      "filename": "KAKAO_898_15.txt",
      "title": "KAKAO_898_15",
      "mediatype": "SNS",
      "medianame": "카카오톡",
      "category": "일상대화",
      "date": "2021-10-05",
      "size": 886,
      "annotations": {
        "subject": "타 국가 이슈",
        "speaker_type": "다자간 대화",
        "size": 886,
        "word_size": 207,
        "text": "1 : 이번에 캘리포니아에 산불 난 거 보셨어요?\n2 : 네 봤어요 ㅜㅜ 일주일 넘게 진압하고 있다는데 걱정이네요...\n3 : 저도 봤어요! 거기 세계에서 가
        "lines": [
```


- **전처리 과정**

1. 데이터 수집

1-1) 파일 불러오기

1-2) 데이터 리스트 만들기

1-3) 텍스트, 주제 불러오기

	text	subject
0	어릴 때 우리 동네에서 땅따먹기 한 거 기억나?	게임
1	키키 땅따먹기?	게임
2	바닥에 그림 그려서 하는 거?	게임
3	응 맞아 키키	게임
4	근데 동네 애기들이 아직도 하더라~	게임
...	...	...
40146	나중에 부산 가면 가봐야겠다 키키	스포츠/레저
40147	응 근데 완전 패키지는 아니래서 궁금해서 갈까 말까 고민 중	스포츠/레저
40148	패키지가 아니면 그냥 입장료만 6만원인 거야?	스포츠/레저
40149	그냥 배 잠깐 빌려 타는 것 같아 키키	스포츠/레저
40150	키키 그럼 그냥 쭈꾸미 낚시 갈래...	스포츠/레저

40151 rows × 2 columns

1-4) 특정 주제(반려동물, 게임 , 연애/결혼) 불러오기

	text	subject
0	어릴 때 우리 동네에서 땅따먹기 한 거 기억나?	게임
1	키키 땅따먹기?	게임
2	바닥에 그림 그려서 하는 거?	게임
3	응 맞아 키키	게임
4	근데 동네 애기들이 아직도 하더라~	게임
...	...	...
40127	근데 그런 거 아까워 하면 끝도 없어 ㅠㅠ	연애/결혼
40128	좋게 생각해야지!	연애/결혼
40129	그렇긴 해!	연애/결혼
40130	요즘 데이트 문제 때문에 현타가 많이 와	연애/결혼
40131	갈수록 현실을 알게 되니까 연애도 힘들고 결혼도 힘든 거 같아 ㅠㅠ	연애/결혼

6043 rows × 2 columns

2. BOW Vectorizer

2-1) 형태소 추출

```
{0: [('어릴', 'Verb'),
  ('때', 'Noun'),
  ('우리', 'Noun'),
  ('동네', 'Noun'),
  ('에서', 'Josa'),
  ('땅', 'Noun'),
  ('따먹기', 'Verb'),
  ('한', 'Verb'),
  ('거', 'Noun'),
  ('기억나', 'Verb'),
  ('?', 'Punctuation')],
 1: [('키키', 'Noun'), ('땅', 'Noun'), ('따먹기', 'Verb'), ('?', 'Punctuation')],
 2: [('바닥', 'Noun'),
  ('에', 'Josa'),
  ('그림', 'Noun'),
  ('그려서', 'Verb'),
  ('하는', 'Verb'),
  ('거', 'Noun'),
  ('?', 'Punctuation')],
 3: [('응', 'Noun'), ('맞아', 'Verb'), ('키키', 'Noun')],
 4: [('근데', 'Adverb'),
```

2-2) Noun, Adjective, verb, adverb추출

2-3) 중복값 제거하기

2-4) BOW데이터프레임 만들기

	어릴	때	우리	동네	땅	따먹기	한	거	기억나	키키	...	먹는데	스무디	시키면	억울하지	않게	헤이즐넛	시럽	소심한	복수	타가
0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0	1.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
40127	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
40128	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
40129	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
40130	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
40131	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

6043 rows × 8150 columns

2-5) Subject칼럼 추가

	어릴	때	우리	동네	땅	따먹기	한	거	기억나	키키	...	스무디	시키면	억울하지	않게	헤이즐넛	시럽	소심한	복수	타가	subject
0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
1	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0	1.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
4	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
40127	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2
40128	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2
40129	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2
40130	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	2
40131	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2

6043 rows × 8151 columns

4. 데이터 저장

최종 데이터 명세

- 독립변수
  - 형태소(형용사, 부사, 명사, 동사)
- 종속변수
  - 0,1,2로 3개로 나눠진 subject 칼럼

IV. 모델링

1. 모델의 정의

- Input: 형태소로 구성된 독립변수와 3개의 값을 가지는 subject 종속 변수
- Output: subject 다중 분류 예측 결과

```
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
```

```
X = word_list_nan.drop(columns = 'subject', axis=1)
y = word_list_nan['subject']
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.3,random_state=0)
```

2. 모델 목록

- RandomForestClassifier

여러개의 결정 트리 분류기가 전체 데이터에서 배깅 방식으로 각자의 데이터를 샘플링해 개별적으로 학습을 수행한 뒤 최종적으로 모든 분류기가 보팅을 통해 예측결정을 하게 된다. 개별 트리가 학습하는 데이터 세트는 전체 데이터에서 일부가 중첩되는 샘플링된 데이터 세트로 부트스트래핑 분할방식을 사용한다.

RandomForestClassifier을 이용해서 다중분류를 진행하였다.

```
rf = RandomForestClassifier(n_estimators=100, random_state=0)
rf.fit(X_train, y_train)
y_pred = rf.predict(X_test)
```

3. 파라미터 튜닝

- GridSearchCV

범위 전체에 대한 모든 조합을 다 진행하여 최적의 파라미터를 찾는다.

```
from sklearn.model_selection import GridSearchCV
params = { 'n_estimators' : [10, 100],
           'max_depth' : [6, 8, 10, 12],
           'min_samples_leaf' : [8, 12, 18],
           'min_samples_split' : [8, 16, 20]
            }
rf = RandomForestClassifier(random_state=0)
grid_cv = GridSearchCV(rf, param_grid = params, cv = 3)
grid_cv.fit(X_train,y_train)
print('최적 하이퍼 파라미터: ', grid_cv.best_params_)
print('최고 예측 정확도: {:.4f}'.format(grid_cv.best_score_))
```

```
최적 하이퍼 파라미터:  {'max_depth': 12, 'min_samples_leaf': 8, 'min_samples_split': 8, 'n_estimators': 100}
최고 예측 정확도: 0.5641
```

⇒ parameter튜닝 결과 성능이 오히려 떨어짐

4. 훈련 결과 및 평가

- 정확도(accuracy)를 이용해서 성능 평가

```
print(f"accuracy_score는 {accuracy_score(y_test, pred)}입니다.")
```

```
accuracy_score는 0.6525096525096525입니다.
```

Ⅴ. 결과 및 요약

AI Hub의 '주제별 텍스트 일상 대화 데이터'를 이용해서 텍스트별 대화 주제를 예측하였다. json파일 형태를 데이터프레임으로 전처리한 후 konlpy와 BOW를 이용해 텍스트 처리를 진행하였고 RandomForestClassifier을 이용해서 주제를 예측해보았다. 정확도를 이용해서 성능을 평가하였고 0.65의 성능을 보였다.

Ⅵ. 피드백

1. 새롭게 학습한 내용

- json파일 전처리를 처음 진행해보았다. 하나의 파일에 정보가 저장되어 있고 여러 이러한 개별 파일을 여러개 이용한다는 것을 알게되었다. 데이터 형태는 중첩된 딕셔너리 형태라는 것을 알게되었고 key값을 이용해서 값을 가져오는 것을 알게되었다.

2. 아쉬운 점

- 파라미터 튜닝을 진행했는데 오히려 성능이 떨어졌다.

3. 다음 분석에서 보완할 것

- 다양한 모델을 사용하고 stacking 및 parametr튜닝 방법을 적용해볼 것이다.

VII. 참고자료

참고자료

- Python-Json-파일-불러오기
- 자동화처리Python-폴더-안의-json파일의-정보를-가져오기

코드, 데이터 첨부

- https://colab.research.google.com/drive/18zzv3gDptdE-CsDBX_j9hbDku6eVWwyr#scrollTo=dUPF5df7tNC4&uniqifier=3