수상작리뷰
[Private 5th] PCA + KMeans&KNN - DACON

[뉴스 기사 레이블 복구 해커톤]

◈ 주제

👉 뉴스데이터를 6개 카테고리로 분류하기. 텍스트 분석

◈ 데이터

- news.csv : 샘플 고유 ID, 뉴스 기사 제목(title), 뉴스 기사 전문(content)
- sample_submission.csv : 샘플 고유 ID , 뉴스 카테고리 (category)

◈ 코드 리뷰

(1) 라이브러리 설치, 임포트, 데이터 로드

- SentenceTransformer 모델 (all-mpnet-base-V2, multi-qa-mpnet-base-dot-v1, all-distilrobert-v1, all-MiniLM-L12-v2, multi-qa-distilbers-cos-v1)

🔷 피처 추가 'text' = title + content

(2) 데이터 전처리

🔷 preprocess_text(text) 함수

- URL, 해시태그, 멘션, 이모티콘, 공백 및 특수문자, 숫자 제거
- text 피처에 적용하여 'processed_text' 피처로 추가

(3) 피처 추출

🔷 SentenceTransformer ('all-distilrobert-v1') 모델 로드

*** 5개 모델들 중에 K-means와 더 적합한 모델 선택***

- 텍스트 피처추출 - encode 메서드

```
sentence_embeddings = model.encode(df['text'].tolist())
```

- 추출한 feature를 데이터프레임에 저장

```
df_embeddings = pd.DataFrame(sentence_embeddings)
```

(4) PCA 차원 축소

🔷 PCA 수행 # df_embeddings 총 768 열 → 중요도 작은 데이터 삭제

- n_components = 0.67, random_state = 64

*** 최적화 파라미터 탐색 방법 - 베이지안 최적화 이론***

- df_embeddings 에 fit_transform → X_reduced

(5) 군집화 및 분류 수행

🔷 **KMeans 수행**

- n_clusters=6, random_state=0
- X_reduced에 fit_predict 하여 'kmeans1_cluster' 피처로 추가

🔷 **KNN 분류 알고리즘 학습**

- n_neighbors=6
- 'kmeans1_cluster' 학습
- 예측 결과를 'knn_cluster' 피처로 추가

⇒ df['kmeans_cluster'] = df['knn_cluster']

(6) Post-processing

🔷df[df['kmeans_cluster'] == 레이블값]['text'].head(3) 으로 확인해서 올바른(정답) 레이블값으로 변환

① Business 0 → 0 (그대로)
② Politics 1 → 2
③ Sport 2 → 3
④ World 3 → 5
⑤ Tech 4 → 4 (그대로)
⑥ Entertainment 5 → 1

(7) Mapping # 전체 데이터에 최종 적용

- mapping_dict = { 0: 0, 1: 2, 2: 3, 3: 5, 4: 4, 5: 1}
- df['mapping'] = df['kmeans_cluster'].apply(lambda x: mapping_dict[x])

◈ **배울점**

- sentence_transformers 라는 텍스트 분석에 사용되는 모델에 대해 알게됨.
- pca 로 차원축소를 한다음 군집화, 분류 까지 이어지는 흐름이 인상깊었음.
- pca 수행 시 최적 파라미터를 베이지안 이론을 활용했다는 점에서 지식의 중요성을 느낌.
- 최종적으로 레이블값을 확인하는 후처리 과정이 필수적임.