Pandas + Seaborn 분석 리포트

일시: 2025년 8월 14일

제출자: 1반 윤소현

목차

1. 데이터 불러오기
2. 결측치 및 기본 정보 확인
3. 기술 통계 및 기본 시각화
4. 범주별 평균 평점
5. 감성 점수와 평점 관계
6. 리뷰 길이와 평점 관계
7. 카테고리 별 평균 감성 점수
8. 인사이트 요약
9. Review_length가 AI 임베딩 유사도에 영향을 줄 수 있나?

1. 데이터 불러오기

```
# 1. 데이터 불러오기
os.chdir("/Users/yshmbid/Documents/home/github/Data-MLOps/0814")

df = pd.read_csv('reviews.csv')
df
```

	review_id	product_id	category	review_text	review_length	num_words	sentiment_score	rating
0	R0001	P158	home	Amazing quality and fast shipping.	134	27	-0.60	3
1	R0002	P117	fashion	Just okay, nothing special.	115	28	-0.10	5
2	R0003	P160	fashion	Not worth the money.	139	32	0.20	5
3	R0004	P127	fashion	Just okay, nothing special.	165	32	0.22	4
4	R0005	P151	home	Just okay, nothing special.	112	15	-0.03	4
...	...	...	...	...	...	...	...	...
195	R0196	P193	electronics	Amazing quality and fast shipping.	131	21	-0.06	5
196	R0197	P153	sports	Not worth the money.	93	13	0.07	5
197	R0198	P169	sports	Excellent product, I loved it!	124	34	-0.29	3
198	R0199	P156	sports	Just okay, nothing special.	121	24	-0.02	1
199	R0200	P194	home	Not worth the money.	85	33	0.39	3

200 rows × 8 columns

- 200개 데이터에 대한 8개 feature가 확인됩니다.

2. 결측치 및 기본 정보 확인

```
# 2. 결측치 및 기본 정보 확인
print("결측치 개수:\n", df.isnull().sum())
print("\n데이터 기본 정보:")
print(df.info())
```

```
결측치 개수:
 review_id          0
product_id         0
category           0
review_text        5
review_length      0
num_words          0
sentiment_score    5
rating             0
dtype: int64

데이터 기본 정보:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 8 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   review_id        200 non-null    object
 1   product_id       200 non-null    object
 2   category         200 non-null    object
 3   review_text      195 non-null    object
 4   review_length    200 non-null    int64
 5   num_words        200 non-null    int64
 6   sentiment_score  195 non-null    float64
 7   rating           200 non-null    int64
dtypes: float64(1), int64(3), object(4)
memory usage: 12.6+ KB
```

- 결측치
  - review text에서 5개, sentimental score에서 5개 결측치가 존재했습니다.
- 결측치 처리에 대한 견해
  - 전체 200행 중 결측치가 단 10행이므로 제거 시 데이터 손실이 5%에 불과하기에 review_text와 sentiment_score 결측인 행을 모두 제거하였습니다.
- 결측치 처리 결과

```
# 결측치 처리: review_text 또는 sentiment_score 중 하나라도 결측인 행 제거
before_rows = df.shape[0]
df = df.dropna(subset=['review_text', 'sentiment_score'], how='any')
after_rows = df.shape[0]

print(f"\n제거된 행 수: {before_rows - after_rows} ({(before_rows - after_rows) / before_rows * 100:.1f}% 데이터 손실)")
print(f"남은 데이터 수: {after_rows}개")
```

```
제거된 행 수: 10 (5.0% 데이터 손실)
남은 데이터 수: 190개
```

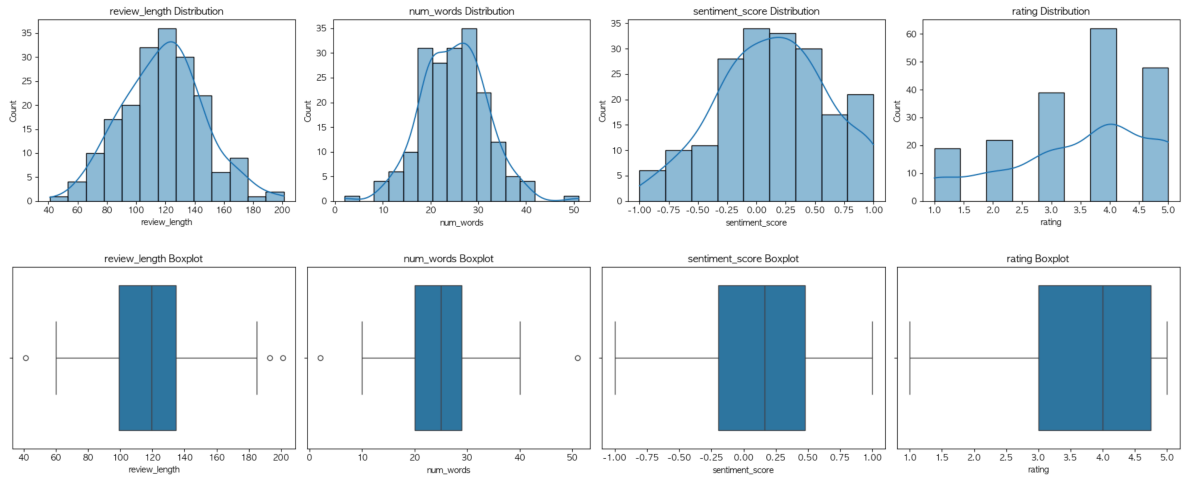
  - 5%에 해당하는 10개 데이터가 제거되어 총 190개 데이터가 분석에 사용되었습니다.

3. 기술 통계 및 기본 시각화

```
# 3. 분포 시각화 및 이상치 탐지
numeric_cols = df.select_dtypes(include=['int64', 'float64']).columns

# 히스토그램
fig, axes = plt.subplots(nrows=1, ncols=len(numeric_cols), figsize=(5*len(numeric_cols), 4))
for ax, col in zip(axes, numeric_cols):
    sns.histplot(df[col], kde=True, ax=ax)
    ax.set_title(f'{col} Distribution')
plt.tight_layout()
plt.show()
```

```
# 박스플롯
fig, axes = plt.subplots(nrows=1, ncols=len(numeric_cols), figsize=(5*len(numeric_cols), 4))
for ax, col in zip(axes, numeric_cols):
    sns.boxplot(x=df[col], ax=ax)
    ax.set_title(f'{col} Boxplot')
plt.tight_layout()
plt.show()
```



- 분포 시각화 결과
  - review_length와 num_words: 종 모양의 분포를 보입니다.
  - sentiment_score: -1에서 1 사이의 범위로 고르게 분포하는데 0 부근에 밀집된 경향이 있고 양쪽 극단값도 있습니다.
  - rating: 4점과 5점이 우세하고 1점과 2점은 적은 비율을 보입니다.
- 이상치 시각화 결과
  - review_length와 num_words: 하단과 상단에 소수의 극단값이 존재합니다.
  - sentiment_score: -1과 1 근처의 극단값이 나타납니다.
  - rating: 1점이 소수의 이상치처럼 보이지만 그냥 낮은 평가일수 있습니다.
- 이상치 처리에 대한 견해
  - review_length와 num_words에 대해서만 이상치를 제거하고 sentiment_score와 rating의 극단값은 데이터의 의미를 유지하도록 이상치처리하였습니다.
  - review_length와 num_words 에 대해서는 상하위 1%를 잘라내는 방식을 사용하였습니다.
- 이상치 처리 결과

```
# 이상치 처리
df_clean = df.copy()

# review_length 상하위 1% 제거
lower_bound = df_clean['review_length'].quantile(0.01)
upper_bound = df_clean['review_length'].quantile(0.99)
df_clean = df_clean[(df_clean['review_length'] >= lower_bound) & (df_clean['review_length'] <= upper_bo
und)]

# num_words 상하위 1% 제거
lower_bound = df_clean['num_words'].quantile(0.01)
upper_bound = df_clean['num_words'].quantile(0.99)
```

```
df_clean = df_clean[(df_clean['num_words'] >= lower_bound) & (df_clean['num_words'] <= upper_boun
d)]

print("상하위 1% 절삭 후 데이터 크기:", df_clean.shape)
```

```
상하위 1% 절삭 후 데이터 크기: (184, 8)
```

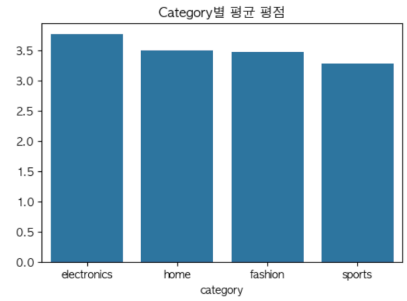
- 1%에 해당하는 6개 데이터가 제거되어 총 184개 데이터가 분석에 사용되었습니다.

4. 범주별 평균 평점

```
# 4. 범주별 평균 평점
category_mean_rating = df.groupby('category')['rating'].mean().sort_values(ascending=False)
print(category_mean_rating)

# 시각화
plt.figure(figsize=(6,4))
sns.barplot(x=category_mean_rating.index, y=category_mean_rating.values)
plt.title("Category별 평균 평점")
plt.show()
```

```
category
electronics    3.764706
home           3.500000
fashion        3.469388
sports         3.285714
Name: rating, dtype: float64
```

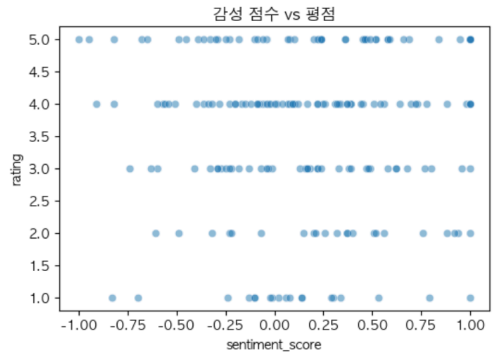


- 범주별 평균 평점 Bar 플롯 확인 결과
  - 네 개 카테고리 모두 평균 평점이 3.2~3.8 사이에 분포해 있습니다.
  - 순위는 electronics, home, fashion, sports 순으로 electronics가 가장 높고 sports가 가장 낮지만 차이는 약 0.5점 정도로 비슷한 편입니다.

5. 감성 점수와 평점 관계

```
# 5. Sentiment Score vs Rating
plt.figure(figsize=(6,4))
sns.scatterplot(data=df, x='sentiment_score', y='rating', alpha=0.5)
plt.title("감성 점수 vs 평점")
plt.show()

# 상관계수 확인
corr_sentiment_rating = df['sentiment_score'].corr(df['rating'])
print("감성 점수와 평점의 상관계수:", corr_sentiment_rating)
```



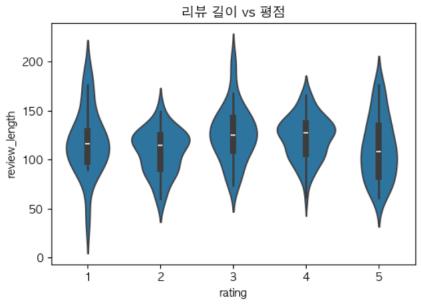
감성 점수와 평점의 상관계수: -0.020926485382556512

- 감성 점수 vs 평점 관계 Scatterplot 확인 결과
  - sentiment_score와 rating의 분포와 경향성: 좌우와 상하로 고르게 분포하며, 특정한 기울기나 뚜렷한 패턴이 나타나지 않았고 이는 두 변수 사이에 양의/음의 관계가 없다고 생각됩니다.
  - 데이터 밀집 구간: sentiment_score가 0 부근에 몰려 있는 경향이 있지만 해당 구간에서도 rating 값이 1부터 5까지 다양하게 나타나 주목할만한 밀집 구간은 없다고 생각됩니다.
  - 이상치: 특정 구간에 극단적으로 높은 또는 낮은 평점이 몰려 있지 않고 전체 범위에서 고르게 분포해 있습니다.
- 상관관계수 확인 결과
  - 상관계수가 -0.0209로 0에 매우 가까우며 감성 점수와 평점 간의 직접적인 상관성을 찾기 어렵다고 생각됩니다.

6. 리뷰 길이와 평점 관계

```
# 6. Review Length vs Rating (violinplot)
plt.figure(figsize=(6,4))
sns.violinplot(data=df, x='rating', y='review_length')
plt.title("리뷰 길이 vs 평점")
plt.show()

# 상관계수 확인
corr_length_rating = df['review_length'].corr(df['rating'])
print("리뷰 길이와 평점의 상관계수:", corr_length_rating)
```



리뷰 길이와 평점의 상관계수: -0.018622392015914393

- 리뷰 길이와 평점 관계 바이올린 플롯 확인 결과
  - 분포 모양: 평점 1부터 5까지 모두 유사한 형태로 퍼져 있고, 특정 평점에서만 길이가 유난히 길거나 짧게 몰리는 현상은 없었습니다.
  - 중앙값: 평점 구간별로 거의 비슷하게 나타나 리뷰 길이의 중심 경향이 평점에 따라 달라지지 않았습니다.
  - 사분위 범위: 평점별로 크게 차이가 나지 않았으며 리뷰 길이 분포가 전반적으로 균질했습니다.

- 극단값 분포: 특정 평점에서만 이상치가 집중되는 현상은 관찰되지 않았습니다.

- 상관 계수 확인 결과
  - 상관계수는 -0.018로 0에 매우 가까워, 선형적인 관계가 없다고 생각됩니다.

7. 카테고리별 평균 감성 점수

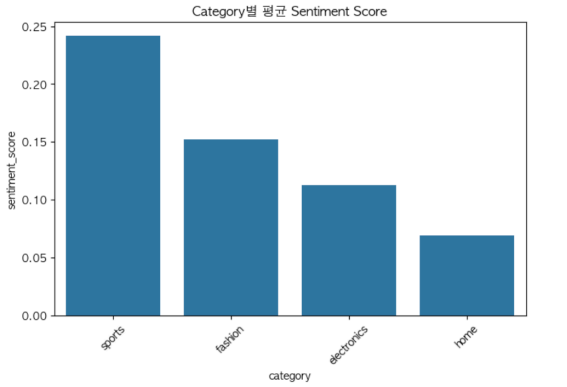
```
# 7. Category별 평균 Sentiment Score
category_sentiment = df_clean.groupby('category')['sentiment_score'].mean().sort_values(ascending=False)
print("Category별 평균 Sentiment Score:")
print(category_sentiment)

plt.figure(figsize=(8, 5))
sns.barplot(data=df_clean, x='category', y='sentiment_score', ci=None, order=category_sentiment.index)
plt.title('Category별 평균 Sentiment Score')
plt.xticks(rotation=45)
plt.show()
```

```
Category별 평균 Sentiment Score:
category
sports         0.241842
fashion        0.152500
electronics    0.112400
home           0.069375
Name: sentiment_score, dtype: float64
/var/folders/5_/cr0lgx655m7fgg_j4q733qbm0000gn/T/ipykernel_65245/2447252410.py:6: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(data=df_clean, x='category', y='sentiment_score', ci=None, order=category_sentiment.index)
```



- 카테고리별 평균 감성 점수 bar plot 확인 결과
  - sports 카테고리가 평균 감성 점수 약 0.24로 상대적으로 긍정적인 리뷰가 많이 기록되었습니다.
  - home 카테고리는 약 0.07로 가장 낮아 다른 카테고리에 비해 긍정적 리뷰가 적었습니다.
  - fashion과 electronics는 각각 약 0.15, 0.11로 중간 수준에 위치합니다.

8. 인사이트 요약

- 감성 점수와 평점의 관계
  - 상관계수가 -0.0209로 0에 매우 가까웠고 scatter plot을 봤을 때도 두 변수 사이에 양의/음의 관계를 확인하기 어렵습니다.
  - 감성 점수와 평점 간의 직접적인 상관성을 찾기 어렵다고 생각됩니다.
- 리뷰 길이와 감성 점수의 관계
  - 상관계수가 -0.018로 0에 매우 가까웠고 violin plot을 봤을 때도 분포 모양, 중심 경향에서 특이점을 찾기 어려웠습니다.
  - 리뷰 길이가 길다고 해서 높은 평점을 주거나 낮은 평점을 주는 경향이 없다고 생각됩니다.
- 카테고리별 감성 차이

- sports 카테고리가 가장 높고 home 카테고리가 가장 낮았습니다.
- sports는 긍정 이미지 강화에 유리한 반면, home은 서비스나 품질 개선 등 부정 요인을 줄이는 전략이 필요하다고 해석됩니다.

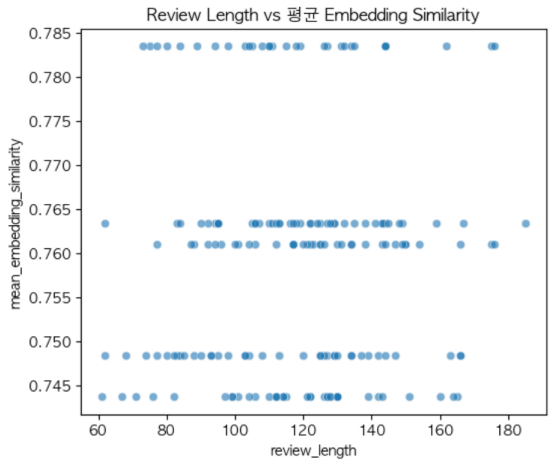
9. Review_length가 AI 임베딩 유사도에 영향을 줄 수 있나?

```
# 9. Review Length vs Embedding Similarity
model = SentenceTransformer('snunlp/KR-SBERT-V40K-klueNLI-augSTS')
texts = df_clean['review_text'].fillna("").tolist()
embeddings = model.encode(texts, convert_to_tensor=True)
reference_embedding = embeddings[0]
similarities = util.cos_sim(reference_embedding, embeddings)[0].cpu().numpy()
df_clean['embedding_similarity'] = similarities

corr_length_similarity = df_clean['review_length'].corr(df_clean['embedding_similarity'])
print(f"Review Length와 Embedding Similarity 상관계수: {corr_length_similarity:.3f}")

plt.figure(figsize=(6, 5))
sns.scatterplot(data=df_clean, x='review_length', y='embedding_similarity', alpha=0.6)
plt.title('Review Length vs Embedding Similarity')
plt.show()
```

```
리뷰 길이와 평균 Embedding Similarity 상관계수: 0.044
```



- 임베딩
  - snunlp/KR-SBERT-V40K-klueNLI-augSTS 임베딩 모델을 사용해 문장을 숫자 벡터로 변환함으로써 각 리뷰 문장을 임베딩하고, 다른 모든 리뷰와의 평균 코사인 유사도를 계산했습니다.
  - 코사인 유사도: 문장의 의미가 비슷하면 벡터 간 코사인 유사도가 높고 다르면 낮아집니다.
- 각 리뷰의 다른 리뷰와의 유사도와 리뷰 길이 간의 관계 확인 결과
  - scatter plot 좌표: (리뷰 길이, 평균 유사도)
  - 리뷰 길이: 60~190 범위에서 고르게 분포합니다.
  - 유사도: 0.745, 0.760, 0.785에 몰려 있습니다.
- 상관계수 확인 결과
  - 상관계수: 0.044
- 인사이트 요약 - Review_length가 AI 임베딩 유사도에 영향을 줄 수 있나?

- 리뷰 임베딩 유사도
  - 유사도 값이 0.745, 0.760, 0.785 부근에 몰려 있어서 리뷰 내용이 몇 가지 반복적인 패턴을 가지는 것 같습니다.
- 리뷰 임베딩 유사도와 리뷰 길이 사이 관계
  - 상관계수는 0.044로 리뷰 길이가 길어질수록 유사도가 조금 증가하는 경향이 있긴 하지만 그 정도가 크지 않다고 생각됩니다.
  - 리뷰 길이가 60~180 사이로 다양하지만 길이에 따라 평균 유사도가 뚜렷하게 변하지는 않으므로 길이가 유사도의 주요 결정 요인이 아니라고 생각됩니다.