네이버 영화 리뷰 분석 프로젝트

감성분석 및 유사도 분석 활용

1조

김용삼, 노지윤, 명소연, 박현주, 정혜리

목차

1. 주제 및 선정 배경

네이버 영화 리뷰 분석 프로젝트
: 감성 분석 및 유사 리뷰 찾기

1. 주제 및 선정 배경

"영화 리뷰의 긍/부정을 예측하고, 유사한 리뷰 추천해주는 서비스"

1

수업 시간에 배운
내용 활용
(긍/부정 예측)

2

주어진 주제들 중에서
가장 흥미로워 보였음

3

영화 리뷰 페이지의
데이터를 활용한
서비스를 구현하고자

2. 프로젝트 결과

웹페이지 구현

2. 프로젝트 결과

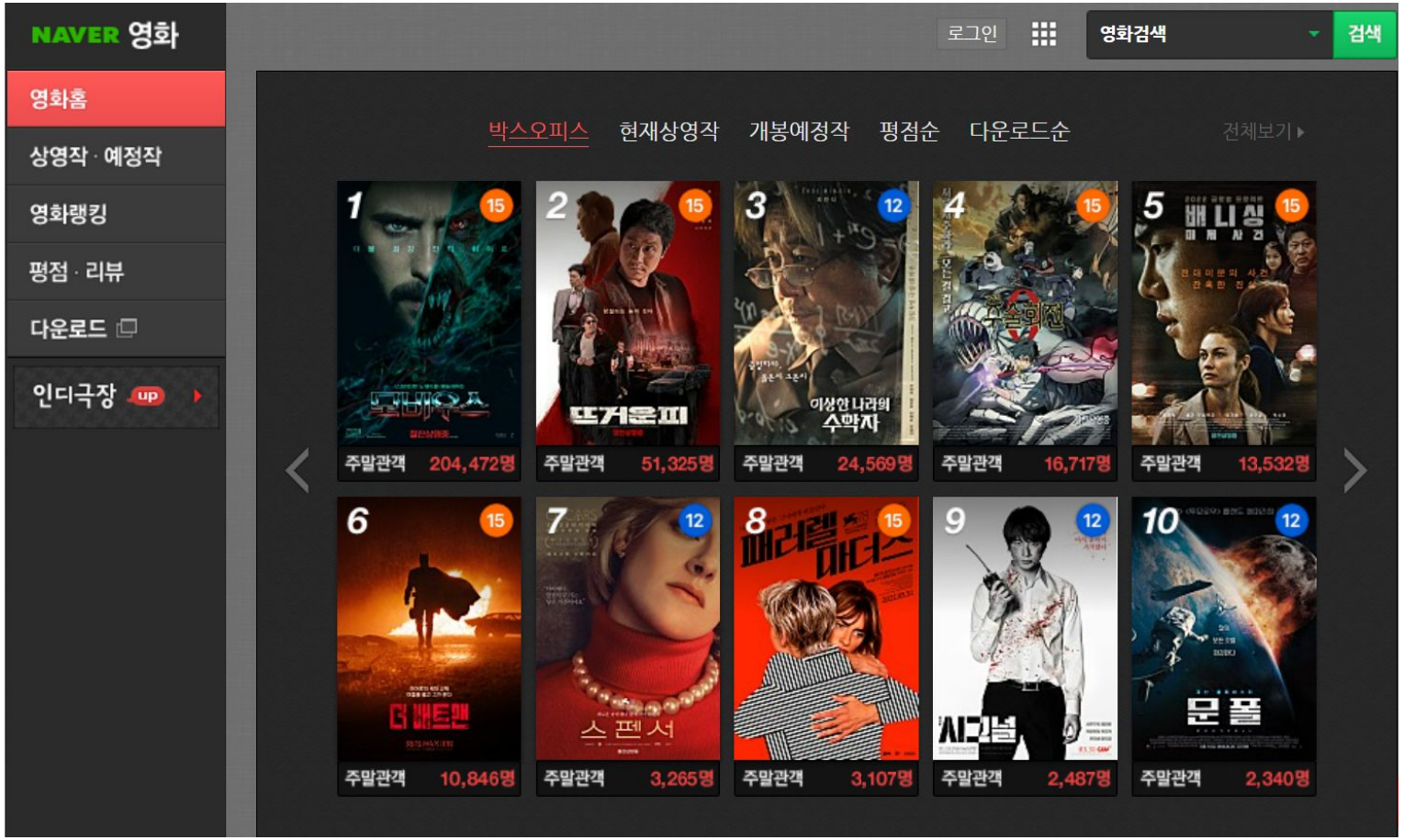
http://54.153.88.29:8000/

github repository

3. 데이터 수집

네이버 영화 리뷰와 평점 데이터

3. 데이터 수집



데이터 수집 사이트: **https://movie.naver.com**

3. 데이터 수집

1) 리뷰 / 평점 페이지

	movie	sentence	score	date	author
0	터미네이터 제니시스	1991년의 상상력을 못이기네	7	22.03.30	gdoq****
1	극장판 금빛 모자이크: 땡큐!!	영화 금빛 모자이크 땡큐 잘 봤습니다.	10	22.03.30	eyyo****
2	파송송 계란탁	아이가 연기를 너무 못하네요. 투박하고 부자연스러웠습니다. 임창정이 연기로 어떻게든...	2	22.03.30	blue****
3	그린랜드	발암 모먼트 탑2. "(군인한테)아까 저희 남편이랑 얘기 하는거 봤어...	1	22.03.30	iblo****
4	모비우스	독약은 두 개 만들고 왜 친구만 보내요ㅜ..	8	22.03.30	bjs6****

총 13,231개의 데이터 수집

3. 데이터 수집

2) 영화 정보 페이지

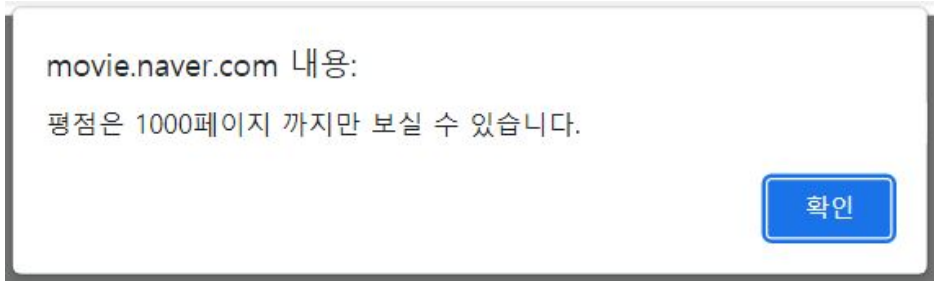
	MOVIE_CODE	MOVIE_TITLE	MOVIE_RATING	MOVIE_GENRE	MOVIE_DIRECTOR	MOVIE_STAR	MOVIE_STORY	MOVIE_RDATE	MOVIE_RTIME
0	10001	시네마 천국	9.63	드라마,멜로/로맨스	쥬세페 토르나토레	마르코 레오나르디,필립 느와레,자끄 페렝,더보기	어린 시절 영화가 세상의 전부였던 소년 토토는 학교 수업을 마치면 마을 광장에 있는...	2020.04.22재개봉,2013.09.26재개봉,1993.11.13재개봉,1990...	124분
1	10002	백 투 더 퓨쳐	9.39	SF,코미디	로버트 저메키스	마이클 J. 폭스,크리스토퍼 로이드,리 톰슨,더보기	힐 밸리(Hill Valley)에 사는 주인공 마티 맥플라이(Marty McFly:...	2015.10.21재개봉,1987.07.17개봉	120분
2	10003	백 투 더 퓨쳐 2	9.65	SF,코미디	로버트 저메키스	마이클 J. 폭스,크리스토퍼 로이드,더보기	브라운 박사(Dr. Emmett 'Doc' L. Brown: 코리스토퍼 로이드 분)...	2015.10.21재개봉,1990.01.13개봉	107분
3	10014	금지된 장난	10.00	드라마,전쟁	르네 클레망	조르주 푸줄리,브리지트 포시,더보기	1940년 6월 남프랑스의 농촌 마을에 파리에서 피난오다 공습으로 부모를 잃고 죽은...	1992.09.26개봉	102분
4	10020	바람과 함께 사라지다	9.47	드라마,전쟁,멜로/로맨스	빅터 플레밍	클라크 게이블,비비안 리,레슬리 하워드,더보기	남북전쟁 발발 직전, 오하라 가문의 장녀 '스칼렛'은 도도한 매력으로 뭇 남성들의 ...	2021.04.28재개봉,1995.05.05재개봉,1972.12.23재개봉,1957...	230분

총 6,823개의 데이터 수집

3. 데이터 수집

3) 사용 데이터셋

- 수집 기간 : **3월 30일 ~ 4월 4일**
- 네이버 영화 평점 페이지의 경우 **1페이지 당 10개의 리뷰를 출력해줌.**
- **1000페이지까지 밖에 제공되지 않음.**



- 추가 데이터셋 : **Naver sentiment movie corpus v1.0** 200,000개

다운로드 링크 : *https://github.com/e9t/nsmc/*

4. 유사도 분석

TF-IDF 벡터화
코사인 유사도

4. 유사도 분석

1) TF-IDF

$$\mathrm{tf}(t,d) = 0.5 + \frac{0.5 \times \mathrm{f}(t,d)}{\max\{\mathrm{f}(w,d) : w \in d\}}$$

TF

(단어 빈도, term frequency)

X

$$\mathrm{idf}(t,D) = \log \frac{|D|}{|\{d \in D : t \in d\}|}$$

IDF

(역문서빈도, inverse document frequency)

사이킷런에서 **TF-IDF**를 자동 계산해주는 **TfidfVectorizer** 사용

```
tfidf = TfidfVectorizer(stop_words=stopwords)
tfidf_matrix = tfidf.fit_transform(df['sentence'])
print('TF-IDF 행렬의 크기(shape) :',tfidf_matrix.shape)
```

```
TF-IDF 행렬의 크기(shape) : (13231, 53612)
```

4. 유사도 분석

2) 코사인 유사도(Cosine Similarity)


```
cosine_sim = cosine_similarity(tfidf_matrix, tfidf_matrix)
print('코사인 유사도 연산 결과 :',cosine_sim.shape)
```

```
코사인 유사도 연산 결과 : (13231, 13231)
```

4. 유사도 분석

3) 분석 결과

```
original_review

movie                    이상한 나라의 수학자
sentence    정말 감동적이다. 배우들의 연기가 너무 좋음
score                                        9
```

리뷰입력

	movie	sentence	score	similarity
2858	스플릿	진짜 감동적이다...ㅜ	10	0.53
5782	뜨거운 피	배우들의 연기가 좋았습니다!!	7	0.41
6235	뜨거운 피	배우들의 연기가 좋았습니다	6	0.41
1058	레버넌트: 죽음에서 돌아온 자	인생영화입니다몇번을봐도 감동적이다	10	0.37
2170	야쿠자와 가족	5번봤다 감동적이다.	10	0.37
8704	킹메이커	배우들의 연기가 대단하다	8	0.35
12950	하로동선	배우들의 연기가 훌륭한 작품	10	0.33
1475	유체이탈자	스토리가 너무 이상함(자기 딱 좋음)	1	0.32
11886	킹메이커	주제를 명확하게 보여주면서 감독 특유의 영상미와 배우들의 연기가 좋음.	8	0.31
9019	클래식	다시 봐도 감동적이다 그때 그 감성을 다시한번	10	0.31

4. 유사도 분석

3) 분석 결과

유사한 리뷰내용 중 빈도가 높은 단어들 워드 클라우드를 통해 시각화

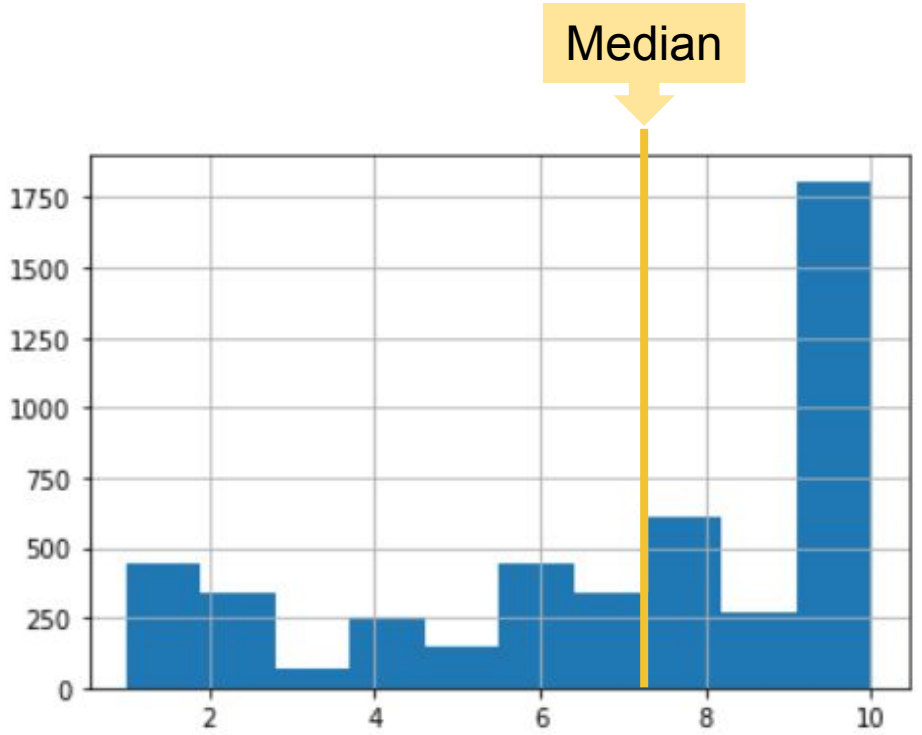
진짜 감동적이다...ㅜ
배우들의 연기가 좋았습니다!!
배우들의 연기가 좋았습니다
인생영화입니다몇번을봐도 감동적이다
5번봤다 감동적이다.
배우들의 연기가 대단하다
배우들의 연기가 훌륭한 작품
스토리가 너무 이상함(자기 딱 좋음)
주제를 명확하게 보여주면서 감독 특유의 영상미와 배우들의 연기가 좋음.
다시 봐도 감동적이다 그때 그 감성을 다시한번

5. 감성 분석

전처리
긍/부정 모델링

5. 감성분석 (전처리)

1) 긍정 / 부정 라벨링



평점 8점을 기준으로 라벨링 한 결과, 긍정과 부정의 비율이 비슷

긍정 (1)
평점 8점 이상

부정 (0)
평점 8점 미만

5. 감성분석 (전처리)

2) py-hanspell 라이브러리

네이버 맞춤법 검사기를 이용한 파이썬용 한글 맞춤법 검사 라이브러리.

```
review['sentence'][20]
```

```
'악의를 가진 특정 1인도 무섭지만,믿고싶은것만 믿는 불특정 다수가 더 무섭다는걸 보여준 영화'
```

```
spelled_sent = spell_checker.check(review['sentence'][20])
hanspell_sent = spelled_sent.checked
hanspell_sent
```

```
'악의를 가진 특정 1인도 무섭지만, 믿고 싶은 것만 믿는 불특정 다수가 더 무섭다는 걸 보여준 영화'
```

5. 감성분석 (전처리)

3) 텍스트 전처리

5. 감성분석 (전처리)

4) 형태소 분석_Okt

konlpy 패키지 중 **Okt 형태소 분석기** 사용하여 **형태소 분석** 및 **태깅 작업**

	sentence	label	tagging
0	믿고 보는 마블 스토리	1	[(믿다, Verb), (보다, Verb), (마블, Noun), (스토리, Noun)]
1	액션을 기대하고 봤다면 반대 연기를 기대하고 봤다면 끝까지	0	[(액션, Noun), (을, Josa), (기대하다, Adjective), (보다...
2	사전 정보 없이 봤는데 러닝타임 내내 시간 가는 줄 모르게 봤습니다	1	[(사전, Noun), (정보, Noun), (없이, Adverb), (보다, Ve...
3	미쳤다 내 시간 순삭 당하고 음	1	[(미치다, Adjective), (내, Noun), (시간, Noun), (순삭,...
4	이 영화는 한마디로 해리 포터제다이스타워즈호빗스타트렉 가지의 영화를 합친 것보다 ...	1	[(이, Noun), (영화, Noun), (는, Josa), (한마디, Noun)...

5. 감성분석 (전처리)

4) 형태소 분석_Okt

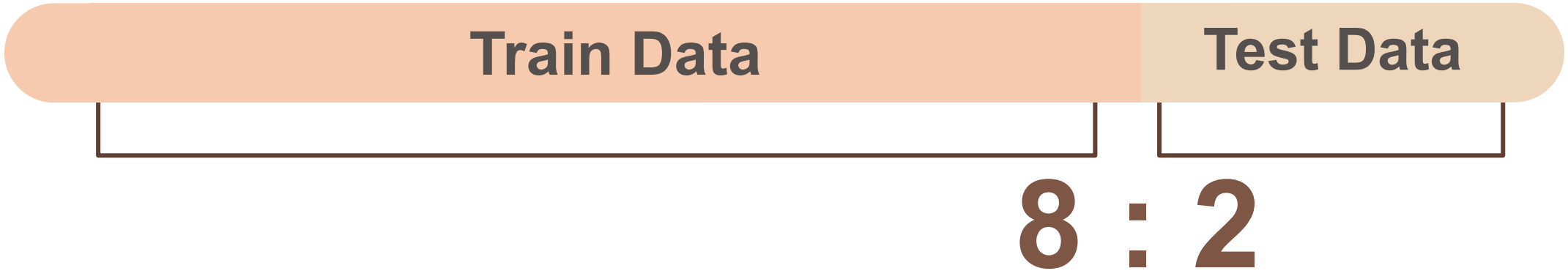
	sentence	label	tagging	tag_list
0	믿고 보는 마블 스토리	1	[(믿다, Verb), (보다, Verb), (마블, Noun), (스토리, Noun)]	[믿다, 보다, 마블, 스토리]
1	액션을 기대하고 봤다면 반대 연기를 기대하고 봤다면 끝까지	0	[(액션, Noun), (을, Josa), (기대하다, Adjective), (보다...	[액션, 기대하다, 보다, 반대, 연기, 기대하다, 보다, 끝]
2	사전 정보 없이 봤는데 러닝타임 내내 시간 가는 줄 모르게 봤습니다	1	[(사전, Noun), (정보, Noun), (없이, Adverb), (보다, Ve...	[사전, 정보, 보다, 러닝, 타임, 내내, 시간, 가다, 줄, 모르다, 보다]
3	미쳤다 내 시간 순삭 당하고 옴	1	[(미치다, Adjective), (내, Noun), (시간, Noun), (순삭,...	[미치다, 내, 시간, 순삭, 당하다, 옴]
4	이 영화는 한마디로 해리 포터제다이스타워즈호빗스타트렉 가지의 영화를 합친 것보다 ...	1	[(이, Noun), (영화, Noun), (는, Josa), (한마디, Noun)...	[이, 영화, 한마디, 해리, 포터, 제다이, 스타워즈, 호빗, 스타트렉, 가지, ...

의미가 있다고 판단되는 **명사, 동사, 형용사**만 추출

5. 감성분석 (모델링)

5) 학습 / 테스트 데이터 분리

전체 데이터셋 : **214,052** 개



5. 감성분석 (모델링)

6) 모델링

CNN

LSTM

CNN
+
LSTM

5. 감성분석 (모델링)

7) CNN

Layer (type)	Output Shape	Param #
embedding_16 (Embedding)	(None, 30, 100)	2534500
conv1d_6 (Conv1D)	(None, 30, 256)	77056
conv1d_7 (Conv1D)	(None, 30, 128)	98432
conv1d_8 (Conv1D)	(None, 30, 64)	24640
max_pooling1d_1 (MaxPooling 1D)	(None, 15, 64)	0
dropout_9 (Dropout)	(None, 15, 64)	0
flatten_1 (Flatten)	(None, 960)	0
dropout_10 (Dropout)	(None, 960)	0
dense_12 (Dense)	(None, 100)	96100
dense_13 (Dense)	(None, 32)	3232
dense_14 (Dense)	(None, 1)	33

Total params: 2,833,993

Conv1D

Dropout 0.25 & 0.3

Dense ‘relu’

Optimizer = ‘rmsprop’

5. 감성분석 (모델링)

7) CNN

```
loaded_model = load_model('review_best_model_10.h5') # 정확도 85.51%
loaded_model.evaluate(X_test, Y_test)
```

```
1298/1298 [==============================] - 10s 8ms/step - loss: 0.3514 - acc: 0.8551

[0.35139697790145874, 0.855140209197998]
```

5. 감성분석 (모델링)

8) LSTM

```
Model: "sequential_10"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_10 (Embedding)    (None, None, 100)         2534500

 lstm_10 (LSTM)              (None, 128)               117248

 dense_9 (Dense)             (None, 1)                 129

=================================================================
Total params: 2,651,877
Trainable params: 2,651,877
Non-trainable params: 0
_________________________________________________________________
```

embedding_dim

vocab_size

hidden_units

Optimizer = ‘rmsprop’

5. 감성분석 (모델링)

8) LSTM

```
loaded_model = load_model('review_best_model_8.h5') # 정확도 85.29%
loaded_model.evaluate(X_test, Y_test)
```

```
1298/1298 [==============================] - 9s 6ms/step - loss: 0.3422 - acc: 0.8529

[0.34223541617393494, 0.8529482483863831]
```

5. 감성분석 (모델링)

9) CNN + LSTM

```
Model: "sequential_4"
_________________________________________________________________
 Layer (type)                 Output Shape              Param #
=================================================================
 embedding_4 (Embedding)      (None, None, 100)         2534500

 conv1d_12 (Conv1D)           (None, None, 128)         64128

 max_pooling1d_4 (MaxPooling  (None, None, 128)         0
 1D)

 conv1d_13 (Conv1D)           (None, None, 128)         82048

 dropout_6 (Dropout)          (None, None, 128)         0

 lstm_2 (LSTM)                (None, 128)               131584

 dense_10 (Dense)             (None, 1)                 129

=================================================================
Total params: 2,812,389
Trainable params: 2,812,389
Non-trainable params: 0
_________________________________________________________________
```

Conv1D

Dropout 0.25 & 0.3

LSTM

Optimizer = ‘rmsprop’

참고논문 : 박호연, & 김경재. (2019). CNN-LSTM 조합모델을 이용한 영화리뷰 감성분석. *지능정보연구*, *25*(4), 141-154.

5. 감성분석 (모델링)

9) CNN + LSTM

```
loaded_model = load_model('review_best_model_15.h5') # 정확도 84.89%
loaded_model.evaluate(X_test, Y_test)
```

```
1298/1298 [==============================] - 5s 4ms/step - loss: 0.3485 - acc: 0.8489

[0.348463237285614, 0.8489257097244263]
```

5. 감성분석 (모델링)

10) 모델 성능 비교

CNN

- Convolution layer 1D :
  256 → 128 → 64
- kernel_size : 3
- MaxPooling1D : pool_size = 2
- Dropout 0.25 → 0.3
- Dense 100 → 32 → 1

CNN

- Convolution layer 1D :
  256 → 128 → 128 →
  64 → 32
- kernel_size : 3
- MaxPooling1D : pool_size = 2
- Dropout 0.25 → 0.3
- Dense 100 → 64 → 32 → 1

LSTM

- LSTM : hidden_units = 128
- Dense 1

CNN

- Convolution layer 1D :
  256 → 128 → 64
- kernel_size : 5
- MaxPooling1D : pool_size = 2
- Dropout 0.3 → 0.3
- Dense 100 → 32 → 1

CNN + LSTM

- Convolution layer 1D :
  128 → 128
- kernel_size : 5
- MaxPooling1D : pool_size = 4
- Dropout 0.25
- LSTM : hidden_units = 128
- Dense 1

Model

- embedding_dim = 100
- hidden_units = 128
- EarlyStopping & ModelCheckpoint
- optimizer = 'rmsprop'
- batch_size = 64
- validation_data = (X_test, Y_test)

5. 감성분석 (모델링)

10) 모델 성능 비교

CNN

- Convolution layer 1D :
  256 → 128 → 64
- kernel_size : 3
- MaxPooling1D : pool_size = 2
- Dropout 0.25 → 0.3
- Dense 100 → 32 → 1

정확도 **85.51%**로 가장 높게 측정됨

5. 감성분석 (모델링)

11) 긍정 / 부정 예측 결과

```
review_predict("솔직히 두 주인공의 감정이입이 안되다보니 아무느낌 없습니다")
```

98.26% 확률로 부정 리뷰

```
review_predict("정겨운 사람과 따뜻한 공간을 떠나온 지금도 가슴 속에 추억한다.")
```

94.33% 확률로 긍정 리뷰

```
review_predict("멸공이라는 단어를 악으로 만든 영화.전형적인 한국식 정치영화")
```

71.79% 확률로 부정 리뷰

6. 마무리하며

기대 효과 및 서비스 발전 방향

6. 기대효과

출력된 리뷰를 통해 사용자에게 맞는 영화를 찾고 추천할 수 있는 서비스 기반

감사합니다.