

썸네일, 영상 제목을 이용한 유튜브 카테고리 분류

# INDEX



- 01** 프로젝트 개요
- 02** 데이터 수집 및 전처리
- 03** 1차 모델 생성/예측/평가
- 04** 특성 추출
- 05** 전체 모델 비교
- 06** 결과 해석



## 01 프로젝트 개요

- 주제 선정 이유
- 프로젝트 진행 순서



## 주제 선정 이유



꽃보다 달리

조회수 14,780회 • 2020. 4. 21.

1.5천 7 공유 7 저장 ...



달려라 달리  
구독자 17.6만명

꽃향기 맡는 것을 참 좋아하는 달리  
예쁜 꽃과 함께 한 귀여운 달리의 모습을 모아봤어요  
모두 힐링하세요~

Bgm  
Patrick Ussher - Song of the Butterflies Strings version  
카테고리 인물/블로그

#로또908회자동 #907회자동 #로또자동  
로또 908회 수요일자동 30장(150게임)

조회수 1,820회 • 2020. 4. 22.



만수르형  
구독자 3.68만명

#로또908회자동 #907회자동 #로또자동  
만수르형밴드 - <https://band.us/n/ada212FeleY3A>  
로또 908회 수요일자동 30장(150게임)

카테고리 인물/블로그

간략히

#해외주식 #디즈니 #애플  
폭발하는 해외 투자와 우리가 사랑하는 글로벌 주식들  
조회수 235,856회 • 2020. 4. 20.



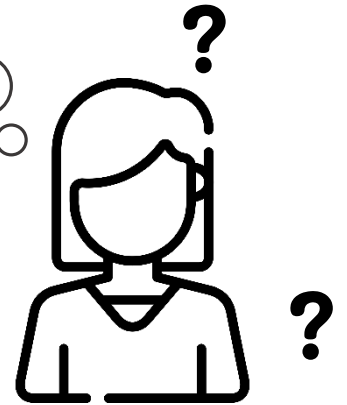
슈카월드  
구독자 65.6만명

어렵고 딱딱한 경제, 시사, 금융 이야기를  
쉽고 유쾌하게 풀어내는  
경제/시사/이슈/잡설 토크방송입니다.

#해외주식 #디즈니 #애플

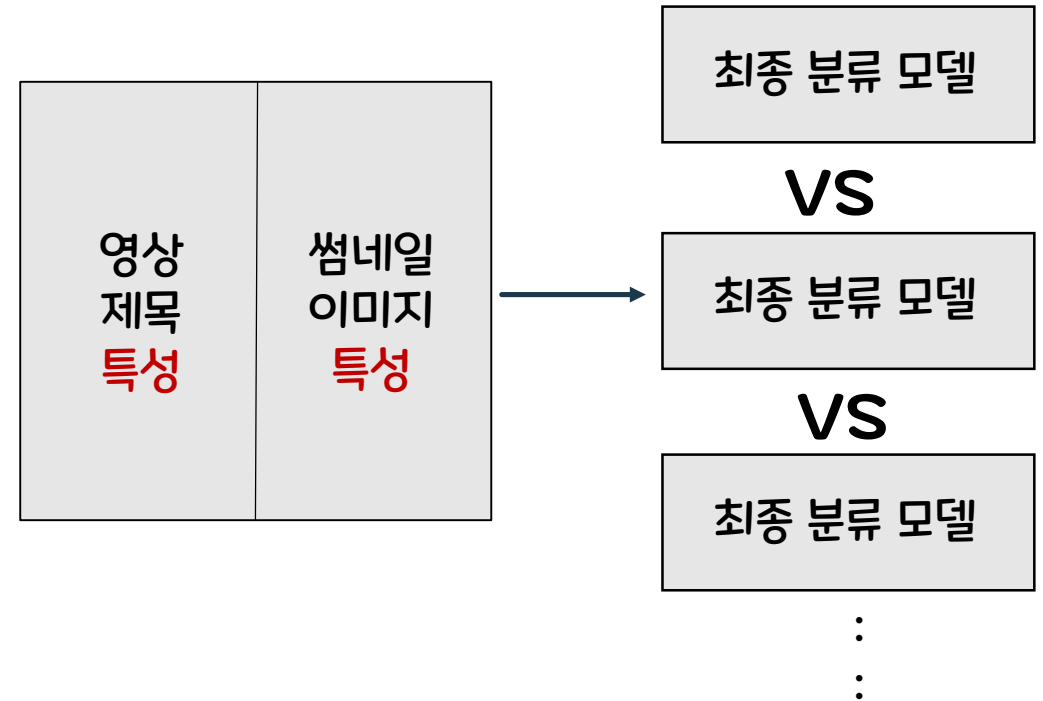
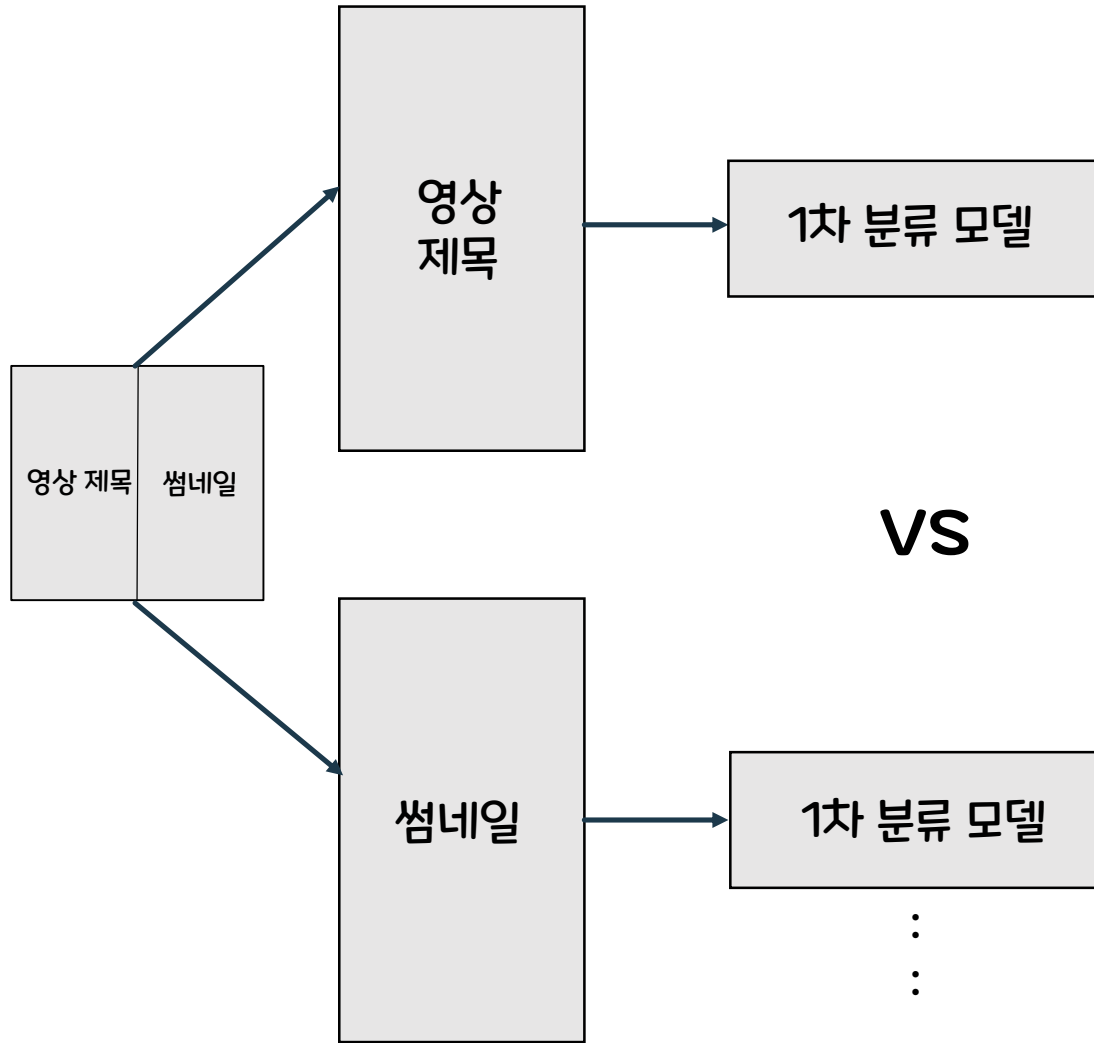
카테고리 코미디

카테고리 분류가  
잘못된 것들이 있네.  
어떻게 하면  
분류 정확도를  
높일 수 있을까?





# 프로젝트 진행 순서





## 02 데이터 수집 및 전처리

- YouTube 데이터 크롤링
- EDA
- 영상 제목 전처리
- 썸네일 이미지 전처리

# Youtube 데이터 크롤링

7개의 카테고리 직접 설정

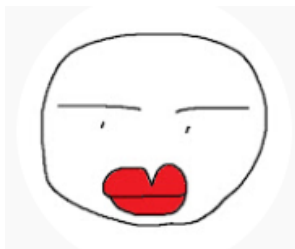
0 : cooking



1 : economy



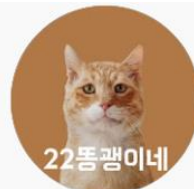
2 : game



3 : movie



4 : pets



5 : politics




6 : sports








# Youtube 데이터 크롤링

 **MochaMilk**  
구독자 101만명


구독중 🔔

홈 동영상 재생목록 커뮤니티 채널 정보 🔍


업로드한 동영상 ▾ 모두 재생 정렬 기준




고양이 장애물 피하기를 강아지가 해보았더니? ㅋㅋㅋㅋ  
조회수 51만회 · 1일 전




강아지는 왜 아기를 지극정성으로 보살폈던걸까? 1 뒤바뀐...  
조회수 117만회 · 5일 전




다시 반년만에 라이브로 인사드려요 😊  
조회수 20만회 · 1주 전




아빠 혼자 놀러갔다고 빠진 강아지  
조회수 93만회 · 1주 전




스테이크 처음 먹어본 강아지! 강아지 스테이크 먹방  
조회수 90만회 · 2주 전  
자막




강제 격리되어 슬픈 강아지의 하루  
조회수 55만회 · 3주 전  
자막




강아지 앞에서 우는척 했다 진짜 울면 했어요..  
조회수 53만회 · 4주 전



사모에드의 유일한 단점, 털! 3일동안 사모에드 털을 모아...  
조회수 324만회 · 1개월 전  
자막



길에서 마주친 엄마가 모른척 지나가자 슬픈 눈빛 보내는 ...  
조회수 68만회 · 1개월 전




강아지에게 갑자기 들이대 불편함을 주어 보았다! 불편한...  
조회수 46만회 · 1개월 전  
자막

2:08 / 10:11

#사모에드 #강아지스파 #나도하고싶다베스랑  
묵묵하는데 이렇게 귀엽기 있기 없기?  
조회수 2,129,142회 · 2019. 1. 20.

4.9만 603 공유 북마크 저장 ...

 **MochaMilk**  
구독자 101만명

구독중 🔔

1) 유튜버의 모든 동영상의 썸네일 이미지 크롤링

2) 동영상의 제목, 채널이름 데이터 크롤링





# Youtube 데이터 크롤링

7개의 카테고리, 전체 13685개의 데이터 수집

|       | channel_name | video_name  | category_id | thumbnail   |
|-------|--------------|---|-------------|---|
| 0     | 복덩이          | 너무나맛있는 '알타리묵은지찜과' "앞치마에그림" Altari old paper st... | 0           | <a href="https://i.ytimg.com/vi/al52wJgMGyl/hqdefault.j...">https://i.ytimg.com/vi/al52wJgMGyl/hqdefault.j...</a> |
| 1     | 복덩이          | 현웃으로만든 [앞치마]와 '민들레' 겹절이 'Apron' made with old ... | 0           | <a href="https://i.ytimg.com/vi/-xx7shznKec/hqdefault.j...">https://i.ytimg.com/vi/-xx7shznKec/hqdefault.j...</a> |
| 2     | 복덩이          | 탈모와다이어트에 탁월하고급진 '느타리버섯볶음' 과 프로방스 창문그리기 Stir-fr... | 0           | <a href="https://i.ytimg.com/vi/m7KC7oEKKuo/hqdefault.j...">https://i.ytimg.com/vi/m7KC7oEKKuo/hqdefault.j...</a> |
| 3     | 복덩이          | 감기 면역력 간기능보호에 탁월한 '도라지튀김' '민들레튀김' '라벤다그리기' Fri... | 0           | <a href="https://i.ytimg.com/vi/FYeuMGEgNWA/hqdefault.j...">https://i.ytimg.com/vi/FYeuMGEgNWA/hqdefault.j...</a> |
| 4     | 복덩이          | 진달래화전 부침 복덩이표 바삭바삭한 쫄쫄독비법공개                       | 0           | <a href="https://i.ytimg.com/vi/UjLKW2fjkyA/hqdefault.j...">https://i.ytimg.com/vi/UjLKW2fjkyA/hqdefault.j...</a> |
| ...   | ...          | ...   | ...         | ...   |
| 13680 | 키다리형         | [VLOG] 체지방 한자릿수 유지식단::뭐든지 과하면 독이다 :: 추억의 VLOG     | 6           | <a href="https://i.ytimg.com/vi/gPEWgwBoT6l/hqdefault.j...">https://i.ytimg.com/vi/gPEWgwBoT6l/hqdefault.j...</a> |
| 13681 | 키다리형         | [키용소 2화] 그들의 입맛을 사로잡은 식단 대공개 (맛있게 다이어트하자)         | 6           | <a href="https://i.ytimg.com/vi/p7Ssy5gB_5A/hqdefault.j...">https://i.ytimg.com/vi/p7Ssy5gB_5A/hqdefault.j...</a> |
| 13682 | 키다리형         | [실속2분팁] 덤벨로우 tip (넓고 두꺼운 등을 위하여)                  | 6           | <a href="https://i.ytimg.com/vi/PI7UN15-5K0/hqdefault.j...">https://i.ytimg.com/vi/PI7UN15-5K0/hqdefault.j...</a> |
| 13683 | 키다리형         | [키용소 1화] 살빼는방법을 모르겠어요 :: 위험한 계약                   | 6           | <a href="https://i.ytimg.com/vi/il4-fi64bss/hqdefault.j...">https://i.ytimg.com/vi/il4-fi64bss/hqdefault.j...</a> |
| 13684 | 키다리형         | [VLOG] 다이어트를 위한 식단과 운동 :: 키용소1기 친구들과 함께한 하루       | 6           | <a href="https://i.ytimg.com/vi/cgp-OT0CEKc/hqdefault.j...">https://i.ytimg.com/vi/cgp-OT0CEKc/hqdefault.j...</a> |

13685 rows × 4 columns



## 영상 제목 전처리

\* 총 4가지의 형태소 분석기 사용

- (1) Okt의 nouns
- (2) Kkma의 nouns
- (3) Soynlp의 word
- (4) Soynlp의 noun

|        | 형태소 분석 방법  | 단점  |
|--------|--|---|
| Okt    | <ul style="list-style-type: none"><li>- 오픈소스 한국어 텍스트 분석기</li><li>- 로딩시간이 상당히 빠름</li></ul>                        | 완전한 수준의 형태소 단위의 분석을 하기 힘들다                          |
| Kkma   | <ul style="list-style-type: none"><li>- 세종 말뭉치를 구조화하고, 이로부터 통계적으로 생성된 말뭉치를 조회</li><li>- 형태소 단위의 분석을 지향</li></ul> | 로딩 시간이 길다   |
| Soynlp | 말뭉치를 cp*에서 학습하고 이를 기반으로 주어진 문장을 토큰화  | 늘 새로운 단어가 만들어지기 때문에 학습하지 못한 단어를 제대로 인식하지 못하는 oov 발생 |

\* Cohesion Probability

연속된 글자의 연관성이 높을수록 단어일 가능성이 높다는 가정 하에 구축된 모델

$$\begin{aligned} cohesion(c_1, c_2, \dots, c_n) &= \sqrt[n]{\prod_{i=1}^{n-1} P(c_1, \dots, c_{i+1} | c_1, \dots, c_i)} \\ &= \sqrt[n]{\frac{Freq(c_1, c_2, \dots, c_n)}{Freq(c_1)}} \end{aligned}$$



## 2. 불용어 사전 정의

### (1) 특수문자



### (3) 영어

#### (4) 숫자

### (1) 코퍼스 빈도어 상위 100개

## (2) 채널명

(3) 눈으로 확인할 수 있는 몇몇 단어



# 영상 제목 전처리

## 3. 명사 단위로 토큰화 (okt, kkma, soynlp)

| 영상 제목   | Okt 형태소 분석기의 명사 단위로 토큰화된 영상 제목           |
|---|--|
| 나이키 축구화 때문에 벌금 1,000만원 낸 김병지 축구화 썰                | [나이키, 축구화, 만원, 김병지, 축구화]                 |
| 매콤곱창볶음   술안주로 제격인 곱창볶음으로 월요일 마무리하기 [만개의레시피]       | [매콤, 곱창, 볶음, 술안주, 곱창, 볶음, 월요일, 마무리, 레시피] |
| 청와대 앞 조국 땀에 난리났다. 조국특검 문정권탄핵 기자회견.(190910)        | [청와대, 조국, 난리, 조국, 특검, 정권, 탄핵, 기자회견]      |
| 총선 D-7, 트로트요정이 투표송을..?😳상큼미 폭발하는 이채윤의 투표송 도전기💪 ... | [총선, 트로트, 정이, 투표, 폭발, 투표, 도전, 정치]        |
| SUB) 귀찮은 그라탕은 이젠 빠엎~!😍😍😍 치즈베이컨그라탕 ★ [만개의레시피]      | [그라탕, 치즈, 베이컨, 그라탕, 레시피]                 |

## 4. 단어 빈도수 계산

|      |      |
|------|------|
| 한국   | 1639 |
| 경제   | 1533 |
| 코로나  | 231  |
| 아이언맨 | 28   |
| ...  | ...  |
| 협찬   | 1    |

단어 중 2번 이상 등장하지 않는  
단어 약 74% 삭제  
약 6000개 단어 사용



# 영상 제목 전처리

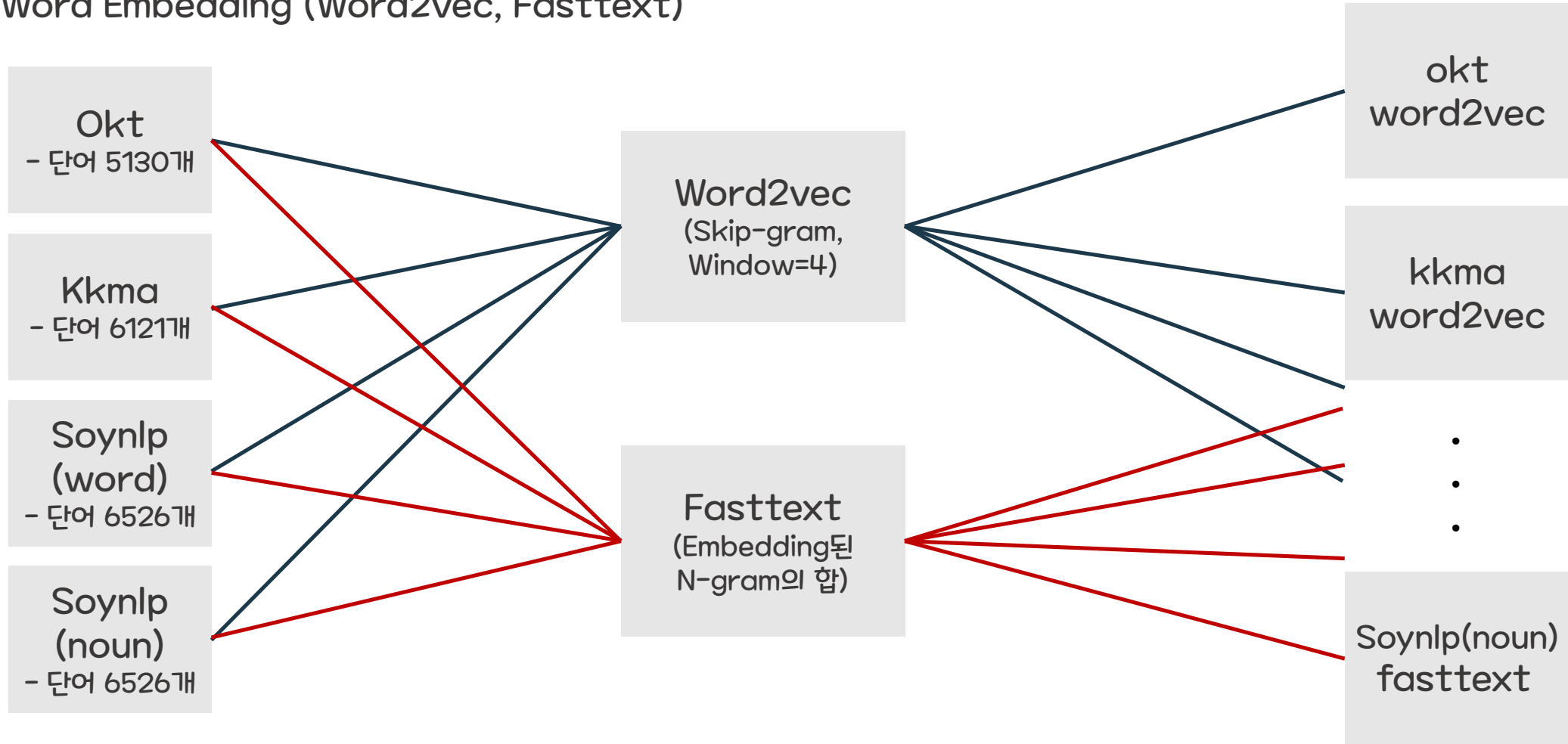
예시 ) 원본 영상 제목과 okt로 명사 토큰화한 제목 비교

| 영상 제목   | Okt로 명사 토큰화한 제목                       |
|---|---------------------------------------|
| 자기 이름 부르면 돌아보는 고백 형제 (ft.오늘 생일🥳)                  | 자기 이름 고백 형제 오늘 생일                     |
| 전 프로 선수의 위플볼 마구! 폭풍삼진쇼                            | 프로 선수 위 볼 마구 폭풍 삼진 쇼                  |
| 좋은에너지를상승시키는법!!!!!!!!!!!!!!                        | 에너지 를 상승 법                            |
| 하늘에서날라온 귀한선물 귀하신손님3                               | 하늘 귀 선물 손님                            |
| 사신이 되어 사람들을 마구 죽이는 게임                             | 마구 죽 게임                               |
| 2018 돈 되는 상가 투자법 제대로 배우기   부동산읽어주는남자              | 돈 상가 투자 법 제대로 부동산 남자                  |
| 🔥 우리집은 김장을 안했지만 생각나는 보쌈과 어울리는 식단 오늘의 식단 🔥 [만개의... | 김장 보쌈 과 식단 오늘 식단 만 개 레시피              |
| 김재원 VS 눈쟁이 배틀그라운드 3만원빵 알까기 대결!                    | 눈 배틀 라운드 만원 빵 대결                      |
| 양파고추장장아찌₩n/양파장아찌관리법₩n&양파엑기스로장아찌활용방법/              | 양파 고추장 장아찌 양파 장아찌 관리 법 양파 로 장아찌 활용 방법 |
| 다시 보는 2차전지 관련주 '코스모신소재'/이희권의 기업탐방 런투유/한국경제TV      | 관련 이희 권 기업 탐방 런투유 한국 경제               |



# 영상 제목 전처리

## 5) Word Embedding (Word2vec, Fasttext)



\* 총 **8가지**의 word Embedding  
1차 성능 평가를 통해 최종 분류 모델에 사용할 형태소 분석기를 정할 것

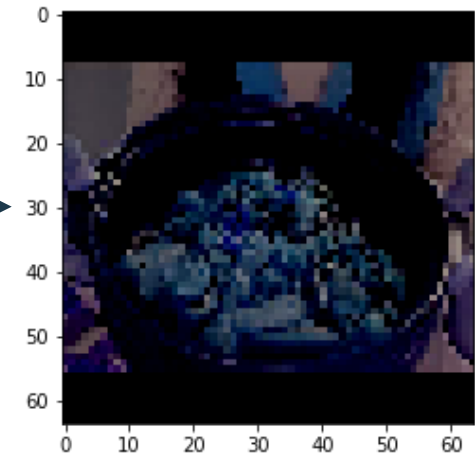
# ▶ 썸네일 전처리

1) 사이즈 변환 및 keras preprocessing하여 컬러 채널 정규화

[https://i.ytimg.com/vi/NhGSdRgqaZ0/hqdefault.jpg?sqp=-ViZkfAF7o8KeHm9HulPg ...](https://i.ytimg.com/vi/NhGSdRgqaZ0/hqdefault.jpg?sqp=-ViZkfAF7o8KeHm9HulPg...)



(360, 480, 3)



(64, 64, 3)





## 03 1차 모델 생성/예측/평가

- 모델 예측 비교 및 평가

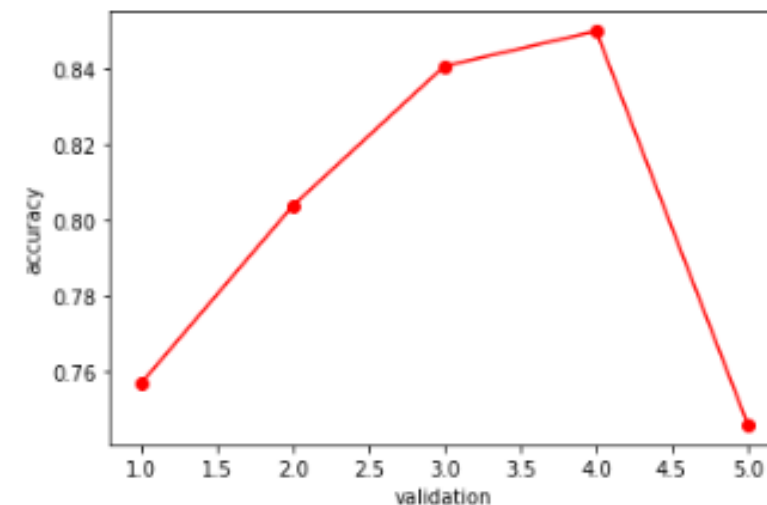


# 모델 예측 비교 및 평가

## 1) 영상 제목만 이용한 분류 모델

|              |          | Accuracy            |          |
|--------------|----------|---------------------|----------|
| 형태소분석기       | 단어 임베딩   | Logistic Regression | SVC      |
| Soynlp(word) | Word2vec | 0.738043            | 0.774467 |
|              | Fasttext | 0.766740            | 0.791391 |
| Soynlp(noun) | Word2vec | 0.741316            | 0.781596 |
|              | Fasttext | 0.78899             | 0.810791 |
| Kkma         | Word2vec | 0.807678            | 0.838317 |
|              | Fasttext | 0.840901            | 0.860465 |
| okt          | Word2vec | 0.825279            | 0.847584 |
|              | Fasttext | 0.843494            | 0.858364 |

- Okt\_fasttext의 LR 정확도 그래프



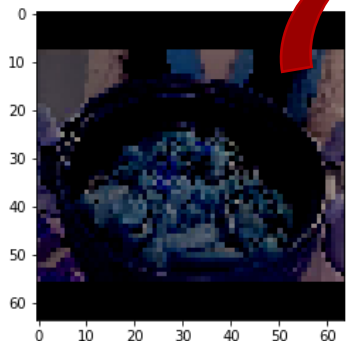
Kkma\_Fasttext 의 SVC 정확도가 가장 높았지만, **kkma의 속도가 매우 느림**

따라서 다음으로 높은 정확도를 보이는 **Okt의 Word2vec, fasttext를 최종 모델 특성추출값으로 사용**

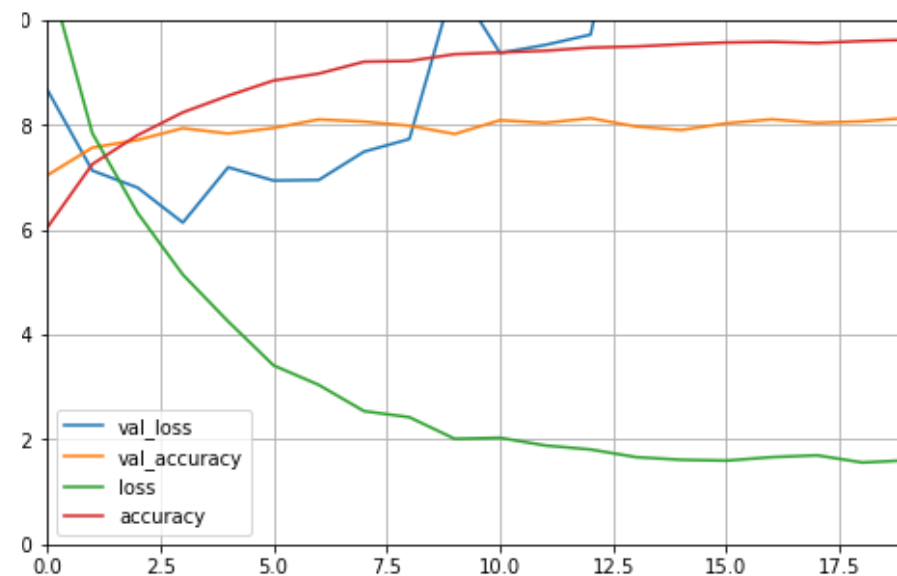
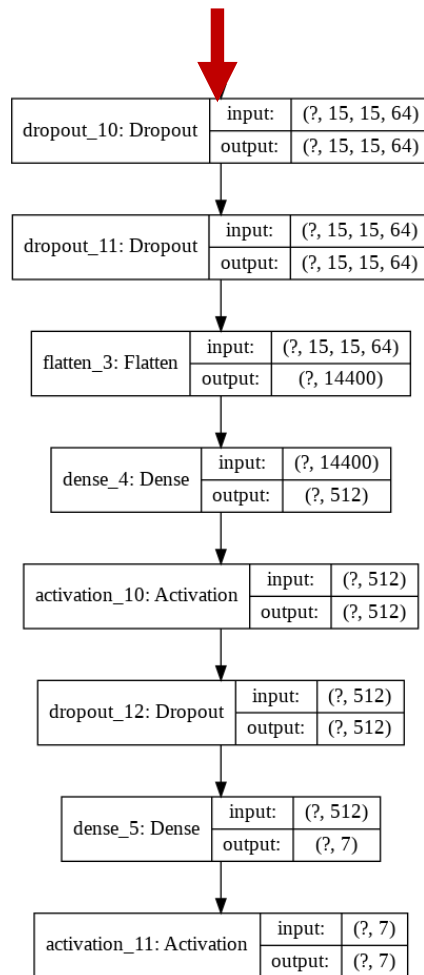
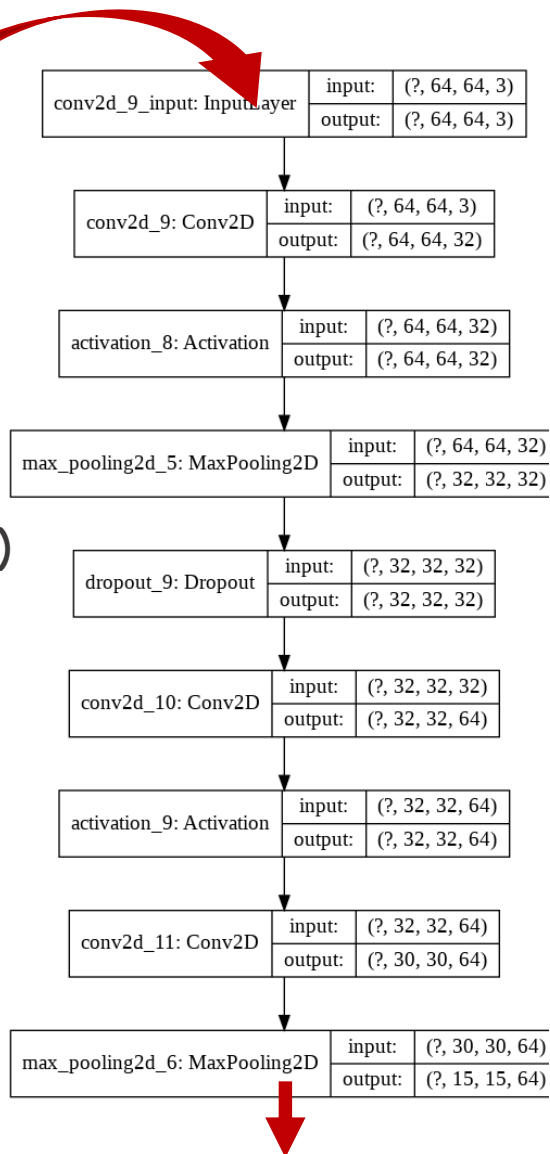


# 모델 예측 비교 및 평가

## 2) 썸네일 이미지만 이용한 분류 모델 – CNN Weight 학습



SHAPE : (64, 64, 3)



- Accuracy = 약 0.81
- Rmsprop loss = 1.4476018495060838

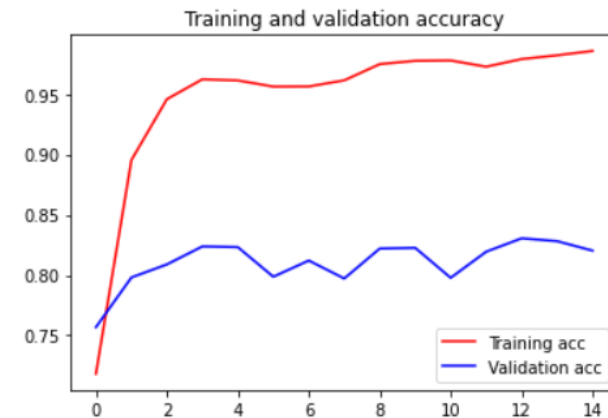
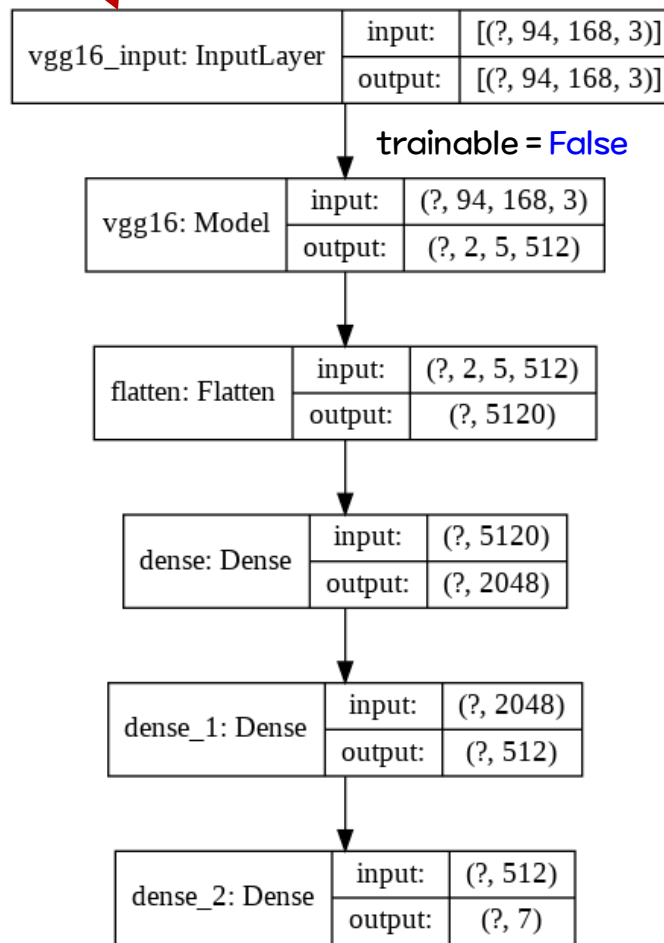


# 모델 예측 비교 및 평가

## 2) 썸네일 이미지만 이용한 분류 모델 – 전이학습 (VGG16)



SHAPE : (94, 168, 3)



- Accuracy = 약 0.82
- rmsprop loss= 1.4476018495060838



## 04 특성 추출

- 영상 제목에서의 특성추출
- 썸네일 이미지에서의 특성추출



# 영상 제목에서의 특성 추출

영상 제목의 Word Embedding 2가지

(1) Okt – Word2vec

|               | V1       | V2       | V3       | ... | V98      | V99       | V100      |
|---------------|----------|----------|----------|-----|----------|-----------|-----------|
| sentence1     | 0.196034 | 0.245908 | 0.576134 | ... | 0.176262 | -0.258607 | 0.123638  |
| sentence2     | 0.198821 | 0.271698 | 0.687552 | ... | 0.23243  | -0.25477  | 0.10881   |
| sentence3     | 0.551256 | 0.671386 | 1.551306 | ... | 0.440023 | -0.734783 | 0.317919  |
| sentence4     | 0.440158 | 0.614208 | 1.510556 | ... | 0.598439 | -0.470475 | 0.210407  |
| sentence5     | 0.359027 | 0.411707 | 0.95849  | ... | 0.303726 | -0.48153  | 0.13332   |
| ...           | ...      | ...      | ...      | ... | ...      | ...       | ...       |
| sentence13621 | 0.183895 | 0.809685 | 0.889106 | ... | 0.386137 | -0.576647 | -0.307054 |
| sentence13622 | 0.581141 | 0.771815 | 1.442716 | ... | 0.387131 | -0.80465  | 0.111101  |
| sentence13623 | 0.233537 | 0.29243  | 0.859342 | ... | 0.406955 | -0.067676 | 0.00364   |
| sentence13624 | 0.323732 | 0.348243 | 0.788682 | ... | 0.258428 | -0.312916 | 0.070189  |
| sentence13625 | 0.579369 | 0.743183 | 1.59592  | ... | 0.423551 | -0.554961 | 0.153877  |

SHAPE : (13625 \* 100)

(2) Okt – Fasttext

|               | V1       | V2        | V3       | ... | V98      | V99      | V100     |
|---------------|----------|-----------|----------|-----|----------|----------|----------|
| sentence1     | 0.523567 | -0.391566 | 0.427402 | ... | 0.636864 | 0.397615 | 0.143179 |
| sentence2     | 0.638376 | -0.498384 | 0.502311 | ... | 0.725406 | 0.380785 | 0.168168 |
| sentence3     | 1.338037 | -1.039847 | 1.181032 | ... | 1.635623 | 0.942006 | 0.454793 |
| sentence4     | 1.318557 | -1.087187 | 1.101964 | ... | 1.54564  | 0.666304 | 0.446539 |
| sentence5     | 0.862312 | -0.558923 | 0.649134 | ... | 1.019527 | 0.61191  | 0.245011 |
| ...           | ...      | ...       | ...      | ... | ...      | ...      | ...      |
| sentence13621 | 0.942522 | -0.647396 | 0.600209 | ... | 1.316312 | 0.435052 | 0.414349 |
| sentence13622 | 1.335089 | -0.878168 | 1.035238 | ... | 1.646191 | 1.131196 | 0.376648 |
| sentence13623 | 0.557477 | -0.804592 | 0.591    | ... | 0.817633 | 0.097795 | 0.260607 |
| sentence13624 | 0.715465 | -0.544113 | 0.499824 | ... | 0.703478 | 0.339621 | 0.180255 |
| sentence13625 | 1.255596 | -1.159646 | 1.080357 | ... | 1.659805 | 0.809986 | 0.246462 |

SHAPE : (13625 \* 100)

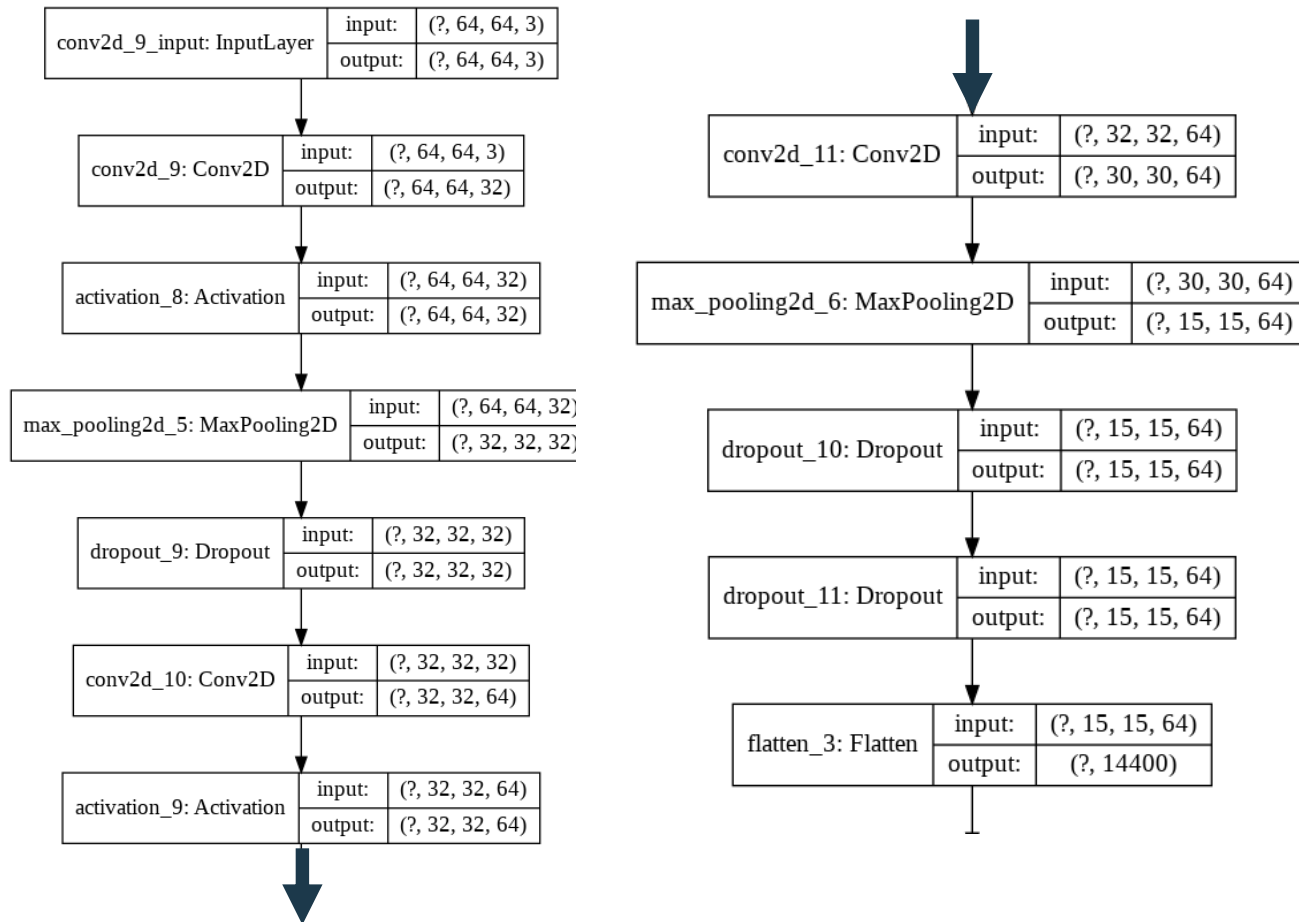
문장의 개수 : 13625개 / 차원 수 : 100



# 썸네일 이미지에서의 특성 추출

## CNN 모델의 Feature Extraction

- Inputs = model, outputs = **model.get\_layer('flatten\_1')** 으로 갖는 flatten\_layer\_model 생성 추출



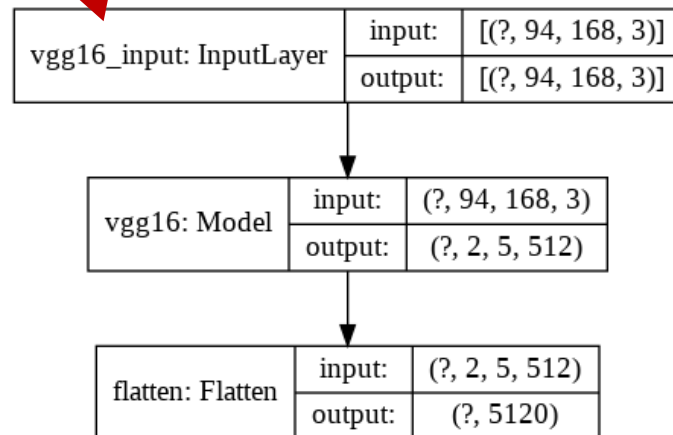
|   | 0         | 1        | 2        | 3         | 4        | 5         | 6         | 7             | 8        |     | 7197      | 7198      | 7199     |
|---|-----------|----------|----------|-----------|----------|-----------|-----------|---------------|----------|-----|-----------|-----------|----------|
| 0 | -0.001158 | 0.002141 | 0.008723 | -0.000788 | 0.004164 | -0.001598 | 0.000216  | -7.286071e-04 | 0.001323 |     | -0.008280 | -0.001530 | 0.001487 |
| 1 | -0.000269 | 0.002502 | 0.008250 | -0.000779 | 0.004414 | -0.001616 | 0.000586  | -6.588997e-07 | 0.000928 |     | -0.006611 | -0.001796 | 0.005069 |
| 2 | -0.001472 | 0.002528 | 0.008565 | -0.001030 | 0.004195 | -0.001770 | 0.000601  | -1.106166e-03 | 0.001294 |     | -0.007296 | -0.001593 | 0.003418 |
| 3 | -0.001088 | 0.002308 | 0.008607 | -0.000768 | 0.004255 | -0.001672 | 0.000101  | -3.708224e-04 | 0.000939 | ... | -0.006626 | -0.001743 | 0.004797 |
| 4 | -0.001340 | 0.002211 | 0.008743 | -0.000668 | 0.004226 | -0.001731 | -0.000070 | -6.905477e-04 | 0.001144 | ... | -0.006554 | -0.001750 | 0.004813 |
| 5 | -0.000788 | 0.002449 | 0.008436 | -0.000797 | 0.004335 | -0.001650 | 0.000430  | -6.423514e-04 | 0.001037 |     | -0.006573 | -0.001754 | 0.004970 |
| 6 | -0.000615 | 0.002377 | 0.008449 | -0.000764 | 0.004301 | -0.001658 | 0.000263  | -2.974104e-04 | 0.000922 |     | -0.006595 | -0.001844 | 0.005117 |
| 7 | -0.000781 | 0.002346 | 0.008618 | -0.000858 | 0.004197 | -0.001719 | 0.000484  | -6.813117e-04 | 0.001280 |     | -0.007401 | -0.001491 | 0.002569 |

(13685, 7200)





## VGG16 모델의 Feature Extraction



|                |                |                |                |
|----------------|----------------|----------------|----------------|
| 0.00000000e+00 | 0.00000000e+00 | 0.00000000e+00 | 0.00000000e+00 |
| 0.00000000e+00 | 6.01975594e+01 | 3.80854106e+00 | 0.00000000e+00 |
| 0.00000000e+00 | 0.00000000e+00 | 2.50367756e+01 | 0.00000000e+00 |
| 0.00000000e+00 | 0.00000000e+00 | 1.73329315e+01 | 0.00000000e+00 |
| 0.00000000e+00 | 0.00000000e+00 | 0.00000000e+00 | 0.00000000e+00 |
| 0.00000000e+00 | 0.00000000e+00 | 0.00000000e+00 | 0.00000000e+00 |
| 0.00000000e+00 | 0.00000000e+00 | 0.00000000e+00 | 0.00000000e+00 |
| 0.00000000e+00 | 0.00000000e+00 | 0.00000000e+00 | 0.00000000e+00 |
| 0.00000000e+00 | 0.00000000e+00 | 0.00000000e+00 | 0.00000000e+00 |
| 0.00000000e+00 | 0.00000000e+00 | 1.87533989e+01 | 0.00000000e+00 |
| 0.00000000e+00 | 0.00000000e+00 | 0.00000000e+00 | 0.00000000e+00 |
| 0.00000000e+00 | 0.00000000e+00 | 0.00000000e+00 | 0.00000000e+00 |
| 0.00000000e+00 | 0.00000000e+00 | 0.00000000e+00 | 0.00000000e+00 |
| 0.00000000e+00 | 0.00000000e+00 | 0.00000000e+00 | 0.00000000e+00 |
| 0.00000000e+00 | 0.00000000e+00 | 0.00000000e+00 | 0.00000000e+00 |

(13685, 5120)



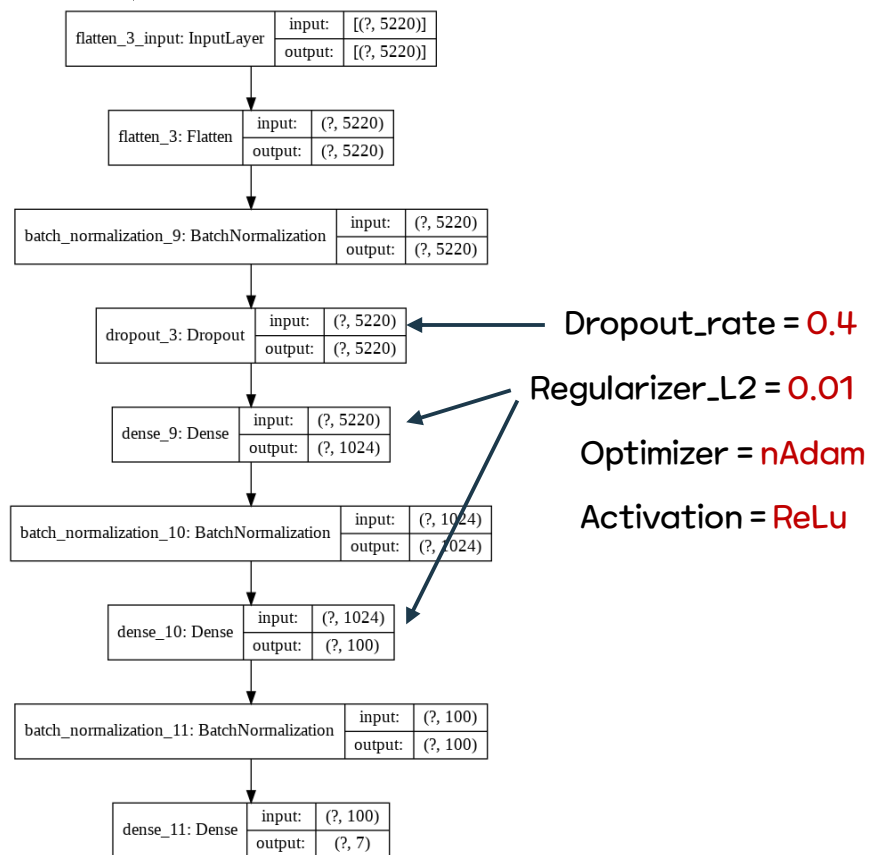
## 05 전체 모델 비교

- VGG16 + Word2vec
- VGG16 + Fasttext
- CNN + Word2vec
- CNN + Fasttext



## “부자가 되기위해 지켜야 할 3가지 원칙 1부 | 부동산업어주는남자”

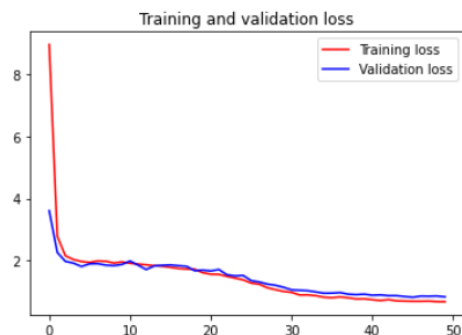
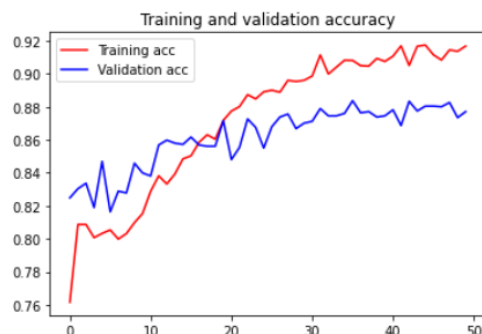
## Word Embedding



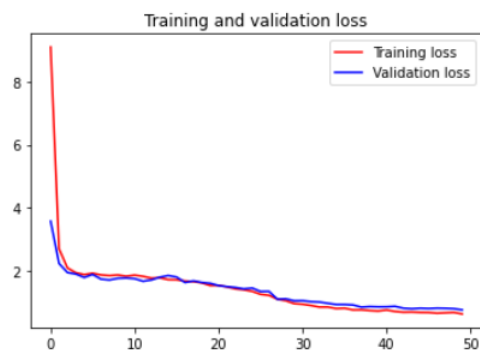
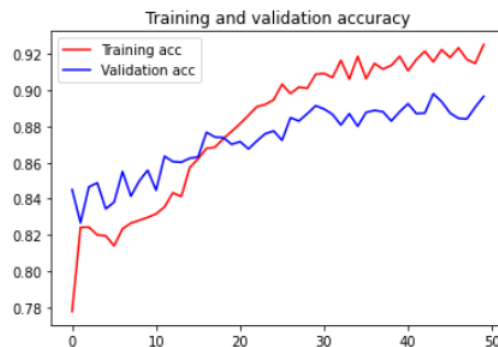


# 전체 모델 비교

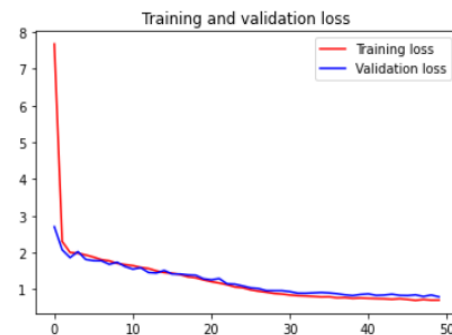
## Model 1. VGG + word2vec



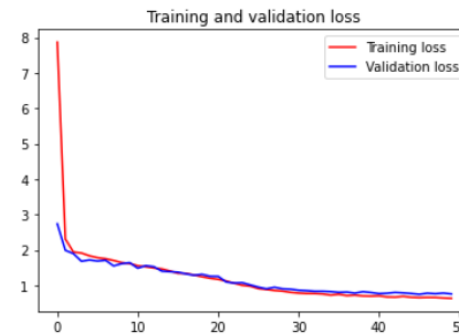
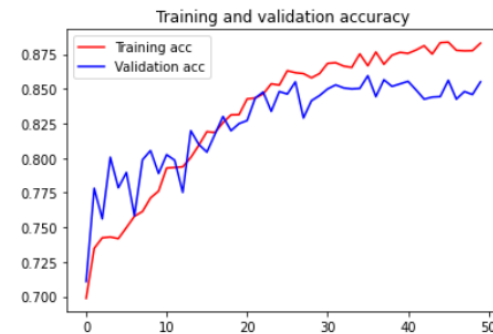
## Model 2. VGG + fasttext



## Model 3. CNN + word2vec



## Model 4. CNN + fasttext





# 전체 모델 비교

## Test

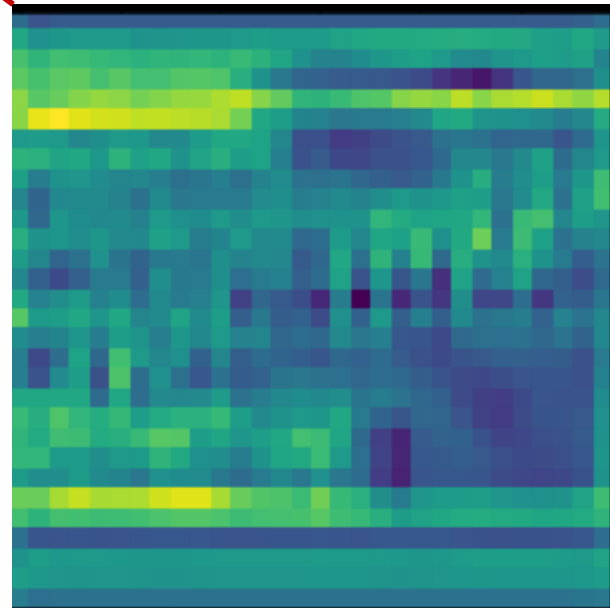
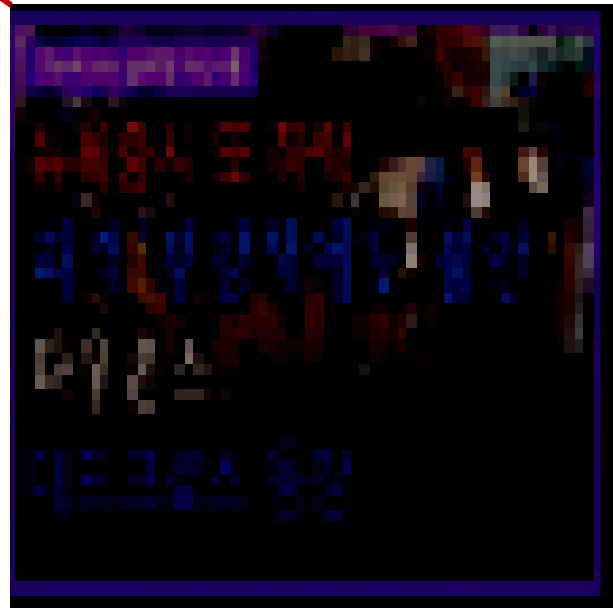
```
[ ] score1 = model1.evaluate(x_case1_test, y_case1_test)
    score2 = model2.evaluate(x_case2_test, y_case2_test)
    score3 = model3.evaluate(x_case3_test, y_case3_test)
    score4 = model4.evaluate(x_case4_test, y_case4_test)

85/85 [=====] - 0s 4ms/step - loss: 0.8520 - accuracy: 0.8752
85/85 [=====] - 0s 4ms/step - loss: 0.7800 - accuracy: 0.8874
85/85 [=====] - 0s 4ms/step - loss: 0.7708 - accuracy: 0.8358
85/85 [=====] - 0s 4ms/step - loss: 0.8205 - accuracy: 0.8370
```

|         | Loss   | Accuracy |
|---------|--------|----------|
| VGG+w2v | 0.852  | 87.52%   |
| VGG+Fxt | 0.78   | 88.74%   |
| CNN+Fxt | 0.7706 | 83.56%   |
| CNN+w2v | 0.8205 | 83.70%   |



## 06 결과 해석







## 결과해석

```
["cooking", "economy", "game", "movie", "pets", "politics", "sports"]
```



Predicted

0

1

2

3

4

5

6

True

0

346

1

12

7

2

1

3

1

1

333

5

0

1

11

3

2

4

3

317

9

3

12

26

3

1

0

18

60

6

6

10

4

16

0

12

8

358

0

8

5

5

3

11

5

3

683

10

6

7

1

37

8

7

20

314



# THANK YOU