기말 프로젝트 보고서

Decoder 팀(최수연, 이준희, 정종원, 송재준, 이길현)

1. 참여한 대회

- "**한국어 문서 추출 AI 경진대회**"에 참여하였다.
- 대회의 주제는 다양한 주제의 한국어 원문으로부터 추출요약문을 도출해낼 수 있도록 하는 인공지능을 개발하는 것이었으며, 구체적으로 말하면 뉴스기사 데이터셋에서 3개의 중요한 문장을 선택하는 추출요약 AI를 개발하는 것이었다.
- 대회기간은 2020년 11월 11일부터 2020년 12월 9일이다.

2. 대회의 평가 지표

- 대회의 평가 지표는 **ROUGE-1, ROUGE-2, ROUGE-L(F1)**이었으며, 각 지표에 대한 순위의 합산 오름차순으로 최종 순위가 결정되었다.
- ROUGE-1과 ROUGE-2는 ROUGE-N에서 N이 각각 1과 2일 때의 지표로, ROUGE-N은 시스템 요약본과 참조 요약본의 문장을 단어 N개씩의 토막으로 분해했을 때 겹치는 단어들의 수를 보는 지표이다.
- ROUGE-L은 LCS기법을 이용해 최장 길이로 매칭되는 문자열을 측정하는 지표이다.

✓ ROUGE (Recall):

$$\frac{\text{number of overlapping n-grams}}{\text{n-grams in reference summary}} = \frac{\sum_{S\in\{\text{ReferenceSummaries}\}} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S\in\{\text{ReferenceSummaries}\}} \sum_{gram_n \in S} Count(gram_n)}$$

✓ ROUGE (Precision):

$$\frac{\text{number of overlapping n-grams}}{\text{n-grams in model summary}} = \frac{\sum_{S\in\{\text{ModelSummaries}\}} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S\in\{\text{ModelSummaries}\}} \sum_{gram_n \in S} Count(gram_n)}$$

✓ **ROUGE (F-1):**

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

3. 프로젝트 수행 과정

- 매주 2회(수요일, 토요일)마다 Zoom 회의를 통해 각자 해 온 결과물을 공유하였으며, 결과를 바탕으로 보완하여 프로젝트를 수행하였다.
- **~11/25**: 각자 맡은 패키지를 실행해본 후 결과 공유한 후 성능이 떨어지는 일부 패키지 drop

: LexRank, GenSim 폐기(종원, 재준) / TextRank, KR-WordRank 채택(수연, 준희, 길현)

- **~11/29**: 데이터 전처리(수연) / Textrank(준희) / KR-WordRank(길현, 종원, 재준)
- **~12/2**: 학습 방식 조정과 Parameter 튜닝을 통해 Rouge 스코어 향상, KR-WordRank 및 TextRank의 한계점에 따른 ML 모델링 고려

: 언론사별 학습, Tokenizer 변경 등(준희) / ML모델링 설계(수연)

- **~12/6**: TextRank 최종 모델 제출, ML 모델링으로 Text Classification 고안

: Text Classification을 위한 데이터 전처리, 피처구축, 모델링 및 파라미터 튜닝(수연, 준희)

4. 데이터 전처리

1) 불필요한 문자 제거

- 뉴스 기사에 특수문자들이 많이 있었고, 일부는 분석에 방해가 되기 때문에 train set에서 실제 요약문에 사용된 특수문자만 제외하고 모두 제거하였다. 이 때 Pattern을 정의하여 영어와 숫자, 한글, 그리고 필수적인 특수문자들만을 제외한 모든 기호를 train set에서 제거하였다.

- Test set에 대해서도 동일한 작업을 수행하였다.

2) 토큰화(Tokenizing)

- 데이터를 알고리즘에 넣기 전 문장을 토큰화하여 단어의 태그(tag)가 명사 (NN), 어근 (XR), 형용사 (VA), 동사 (VV)인 것만 추출하였다.

- NLP(자연어 처리) 분야에서 일반적으로 가장 우수한 성능을 내는 Tokenizer인 Mecab을 적용하였다.

5. KR-WordRank (한국어의 어절 구조를 고려한 비지도학습 알고리즘)

1) 알고리즘의 적용 순서

① 어절의 위치(L,R)에 따라 substring의 ranking을 계산

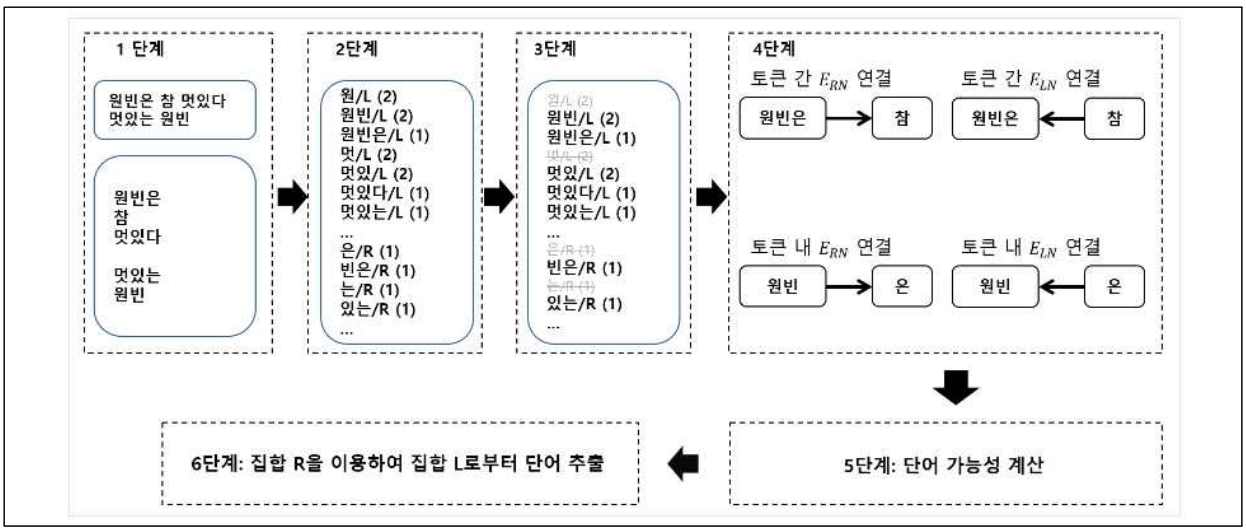
- 여기서 L, R은 한국어 어절 구조의 특징 때문에 만든 것이며, L은 어절의 왼쪽에 위치한 글자를, R은 어절의 오른쪽에 위치한 글자를 의미한다.

② 같은 위치(L,R)이면서 빈도수가 같은 subsrtings 제거

③ 어절 내 L 과 R 와 어절 간 링크를 구성

④ Graph ranking 학습

⑤ R 중 상위 rank를 suffix set 으로 선택 후 rank 기준으로 L 을 필터링하여 keywords 추출



2) 파라미터 튜닝

- 성능 향상을 위해 KR-WordRank 패키지 내의 파라미터 값을 일부 변경하였다. 10~50개 가량의 단어로 이루어진 문장을 선호도록 하고, Keywords 수를 10개로, stopwords에 조사인 '이다, 였다, 했다'를 추가하였고, diversity값을 0.8로 변경하였다.(여기서 diversity는 문장을 추출할 때 추출한 문장들이 비슷한 의미를 갖고 있도록 하지 않게 하는 파라미터로, diversity의 값을 높게 할수록 비슷한 의미의 문장이 덜 추출되도록 한다.)

- 파라미터 튜닝 결과 ROUGE-L 점수가 0.276에서 0.278로 소폭 상승하였으나, 파라미터를 더욱 다양하게 변경하더라도 크게 성능이 향상되지 않아 KR-WordRank 패키지는 이 정도로만 탐구하였다.

6. TextRank (PageRank 기반 비지도학습 알고리즘)

1) 알고리즘 적용 순서

① 핵심 단어를 선택하기 위해 단어 간의 co-occurrence graph 생성

② 핵심 문장을 선택하기 위해 문장 간 유사도 기반 sentence similarity graph 생성

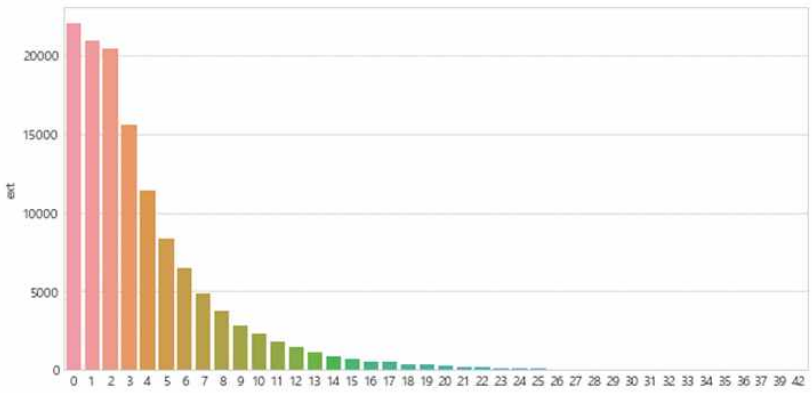
③ 각각 graph에 PageRank[1]를 학습하여 각 마디(단어 혹은 문장) 의 랭킹 계산

$$TR(V_i) = (1-d) + d * \sum_{V_j \in In(V_i)} \frac{W_{ji}}{\sum_{V_k \in Out(V_j)} W_{jk}} TR(V_j)$$

- $TR(V_i)$: 문장 또는 단어(V)i에 대한 TextRank값
- W_{ij} : 문장 또는 단어 i 와 j 사이의 가중치
- d : dampingf actor, PageRank에서 웹 서핑을 하는 사람이 해당 페이지를 만족하지 못하고 다른페이지로 이동하는 확률로써, TextRank에서도 그 값을 그대로 사용(0.85로 설정)
- TextRank $TR(V_i)$를 계산 한 뒤 높은 순으로 정렬

2) Bias Feature

	length
count	42803.000000
mean	13.256594
std	5.965359
min	3.000000
25%	9.000000
50%	12.000000
75%	16.000000
max	97.000000



- 주어진 train set에서, 기사들은 3분위수까지가 20문장 이하로 이루어져 있음을 확인하였다.
- 또한 주어진 train set의 'extractive' 변수는 몇 번째 문장이 요약에 선택되었는지를 나타내는 변수로, 이 변수의 분포를 확인했을 때 모든 기사의 문장들이 동일한 확률로 선택되지 않았음을 확인할 수 있었다. 오히려 기사의 앞쪽에서 실제 사람이 요약한 요약문에 선택된 문장이 몰려있음을 알 수 있었다.
- 따라서 이러한 이유로 TextRank 패키지에는 들어있지 않은 **'bias'라는 새로운 피처**를 자체적으로 만들었다.

3) Media Feature

- 주어진 train set에서의 '**media**' 변수는 기사를 발행한 언론사를 나타내는 변수이다. 처음에는 별 의미 없는 변수라고 생각하였으나, TextRank 알고리즘을 media별로 돌리면 ROUGE 점수가 현저하게 달라지는 것을 확인할 수 있었다. 그래서 media별로 파라미터 튜닝을 하도록 하였다.

4) 하이퍼 파라미터 튜닝

- TextRank에서의 하이퍼 파라미터는 min_sim, dumping factor 2가지인데, 앞서 이야기한 대로 media별로 train set을 나누어 2개의 하이퍼 파라미터를 튜닝하여 ROUGE-F1(=ROUGE-L) 점수를 최대로 만들도록 진행하였다.
- 여기서 min_sim은 TextRank에서 word occurrence graph를 만들 때 min_sim 이상의 단어들만 고려하도록 하는 파라미터이고, dumping factor는 웹서핑을 하는 사람이 해당 페이지를 만족하지 못하고 다른 페이지로 이동하는 확률로, 디폴트로는 0.85를 사용한다.

---

1) PageRank란 구글에서 개발한 초기 검색 엔진 알고리즘으로, 자세한 내용은 논문 "Authoritative Sources in a Hyperlinked Environment. Jon M. Kleinberg"를 참고.

5) 주의할 점

- TextRank 알고리즘에서는 validation set을 따로 만들지 않았다. 왜냐하면 TextRank는 실제 요약문을 학습시키지 못한다는 치명적인 한계점이 있었기 때문이다. 우리 팀은 TextRank의 하이퍼 파라미터인 min_sim, dumping factor 등의 조정을 통해 정해진 TextRank 알고리즘 하에서 최대의 성능을 내도록 만들었으나, 최대 성능에는 한계가 있었다.

- TextRank와 KR-WordRank를 이용해 test set에 대한 ROUGE 점수는 다음과 같다. 최종 제출물로는 TextRank를 선택하였고, 총 392팀 중 23등의 순위를 기록하였다.

패키지	TextRank	KR-WordRank
Rouge-1(점수1)	0.458	0.388
Rouge-2(점수2)	0.312	0.222
Rouge-L(점수3)	0.453	0.278
최종 선택	O	X

7. Text Classification

- 공모전이 종료되었지만 TextRank의 성능에 아쉬움을 느꼈던 우리 팀은 더 좋은 성능을 내는 다른 추출요약 알고리즘들을 찾아보았다. 이 과정은 쉽지 않았는데, 현대의 Text Summarization에 해당하는 SOTA(State Of The Art) Model은 Bert 계열의 모델이라 이를 활용하고 싶었지만 모델 하나를 학습시키는 데에만 최소 며칠에서 1주일 이상의 시간이 소요되었기 때문에 우리 조의 컴퓨팅 자원으로는 이를 활용할 수 없었다.

- 그래서 새롭게 찾아낸 기법이 바로 **머신 러닝 알고리즘을 활용한 Text Classification**(텍스트 분류)였다. 우리는 여러 논문들을 찾아 이 기법을 우리 데이터에 적용시켰다.

1) 선행 연구 고찰

- 참고한 논문은 총 3 가지로, 각각 'Automatic Text Summarization Using a Machine Learning Approach'[2], 'Free Model of Sentence Classifier for Automatic Extraction of Topic Sentences'[3], 'Document Clustering and Text Summarization'[4]이다. 이 중 첫 번째 논문을 가장 많이 참고하였다.
- 위 논문들은 모두 텍스트 분류에 관한 것으로, 텍스트의 각 문장 중 선택된 문장을 1, 선택되지 않은 문장을 0 으로 labeling 한 후 텍스트의 여러 가지 특징들을 Feature 로 만들고, 머신 러닝 모델에 적용하는 방식을 활용하였다. 논문에서 텍스트 분류의 머신 러닝 모델로 Naive Bayes, Decision Tree Classifier 의 성능이 좋다고 언급한 점을 참고하여, 우리 프로젝트에서도 이를 활용하였다.

2) 데이터 전처리

- 우리 데이터의 경우는 텍스트가 수만 개의 기사로 되어 있어 각 기사의 문장들을 뽑고 피처를 만든 후 train set 과 validation set 으로 나누면 모든 기사들이 섞이는 문제가 발생하였다. 이 때문에 각 기사의 개별 id 를 기준으로, 주어진 train 데이터에 대해 기사별로 먼저 train set 과 validation set 으로 나누었으며, 그 비율은 8:2 로 하였다.
- 나뉜 train set 과 validation set 에 대해, 기사 원문이 들어 있는 'article_original' 변수에 대해 각 기사를 문장으로 나누고, 이를 '**article_sent**'라는 변수로 명명하였다. 그리고 'extractive' 변수에 있는 선택된 문장 번호를 기준으로 어떤 문장이 선택되었는지를 0 과 1 의 값으로 labeling 한 뒤 이 변수는 **'label'**로 명명하였고, 이를 target 변수로 하여 예측하였다.
- 다음으로는 'article_sent'의 문장 내의 불필요한 문자들을 앞의 '4. 데이터 전처리'에서 정의한 pattern을 이용해 정제하였다. 전처리가 완료된 결과는 다음과 같다.

---

2) Automatic Text Summarization Using a Machine Learning Approach. Joel Larocca Neto; Alex A. Freitas; Celso A. A. Kaestner. 2002.
3) Free Model of Sentence Classifier for Automatic Extraction of Topic Sentences. M.L. Khodra; D.H. 2011.
4) Document Clustering and Text Summarization. Larocca Neto, J.; Santos, A. D.; Kaestner, C.A.; Freitas, A.A. 2000.

	media	id	article_original	abstractive	extractive
0	당진시대	327827480	[당진시 문화관광과를 대상으로 하는 행정사무감사에서 당진시립합창단 관계자가 보낸 것...	지난 6일 당진시의회 행정사무감사에서 '합창단이 소리를 작게 낼 것이니 알고 있으라...	[0, 1, 2]
1	국제신문	339840364	[미국 메이저리그(MLB)에서 활동하는 한국 선수들의 시즌 초반 희비가 엇갈리고 있...	LA 에인절스의 최지만이 맹활약을 하여 시즌 타율 0.250에서 0.313으로 올리...	[0, 1, 3]
2	기호일보	371071597	[인천 부영공원 운영 생활야구협회 80여 동호회 팀에 260만 원씩 받아, 국유지로...	16일 부평구와 협회 등에 따르면 부영공원 안에 있는 야구장을 구생활체육야구협회가 ...	[6, 7, 10]
3	대구일보	354806783	[대구·경북첨단의료산업진흥재단 의약생산센터는 항암주사제 무균충전 시설을 갖추고 있다...	대구·경북첨단의료산업진흥재단 의약생산센터는 약사법 시행규칙에서 정한 바에 따라 전용...	[1, 6, 4]
4	대구신문	347022773	[국내 유통되는 탄산음료 중 식품의약품안전처에 품질인증을 받은 제품이 하나도 없는 ...	식품의약품안전처는 29일 어린이가 즐겨마시는 음료를 대상으로 영양성분을 조사한 결과...	[2, 3, 10]

▲ 주어진 train 데이터(원본)

	media	article_original	abstractive	extractive	id	article_sent	label
0	국제신문	[부산시가 2032년 하계올림픽 국내 유치 신청 도시 선정을 앞두고 경쟁 도시인 서...	부산시가 2032년 하계올림픽 국내 유치 신청 도시 선정을 앞두고 경쟁 도시인 서울...	[0, 5, 3]	333727228	부산시가 2032년 하계올림픽 국내 유치 신청 도시 선정을 앞두고 경쟁 도시인 서울...	1
1	국제신문	[부산시가 2032년 하계올림픽 국내 유치 신청 도시 선정을 앞두고 경쟁 도시인 서...	부산시가 2032년 하계올림픽 국내 유치 신청 도시 선정을 앞두고 경쟁 도시인 서울...	[0, 5, 3]	333727228	오거돈 부산시장은 11일 오후 충북 진천 국가대표 선수촌에서 열릴 대한체육회의 대의...	0
2	국제신문	[부산시가 2032년 하계올림픽 국내 유치 신청 도시 선정을 앞두고 경쟁 도시인 서...	부산시가 2032년 하계올림픽 국내 유치 신청 도시 선정을 앞두고 경쟁 도시인 서울...	[0, 5, 3]	333727228	대의원 총회 심의에는 오거돈 부산시장과 박원순 서울시장이 참석해 각각 올림픽 유치 ...	0
3	국제신문	[부산시가 2032년 하계올림픽 국내 유치 신청 도시 선정을 앞두고 경쟁 도시인 서...	부산시가 2032년 하계올림픽 국내 유치 신청 도시 선정을 앞두고 경쟁 도시인 서울...	[0, 5, 3]	333727228	두 광역단체장의 발표와 질의응답을 거친 뒤 대의원 투표를 통해 국내 유치 도시가 확...	1
4	국제신문	[부산시가 2032년 하계올림픽 국내 유치 신청 도시 선정을 앞두고 경쟁 도시인 서...	부산시가 2032년 하계올림픽 국내 유치 신청 도시 선정을 앞두고 경쟁 도시인 서울...	[0, 5, 3]	333727228	오 시장은 한반도는 되돌릴 수 없는 평화의 미래를 향해 또 한 번 큰 걸음을 내딛는...	0

▲ 각 'article_original' 변수의 기사 원문을 문장별로 나눠 정리한 'article_sent' 변수 생성

3) 피처 구축

- 우리 데이터에서 만든 피처는 총 7가지로, 논문을 참고하여 만든 피처도 있고 우리 데이터에 맞게 자체적으로 구성한 피처도 있다.

(1) **SL**(Sentence Length): 'article_sent' 변수에 있는 문장을 토큰화하여 필요한 토큰들(명사 (NN), 어근 (XR), 형용사 (VA), 동사 (VV)인 것)만 추출해 'article_token'에 저장하였다. 여기서 각 문장의 token 개수를 반환하고, 각 기사 내에서 가장 긴 문장의 token 개수를 반환하여 두 개의 비(ratio)를 계산하여 이 피처를 만들었다.

(2) **TF-ISF**: TF(Term Frequency)와 ISF(Inverse Sentence Frequency) 값을 곱한 값이다. 먼저 TF(s,t)는 특정 문장 s에서의 특정 단어 t의 등장 횟수이고, ISF(t) 값은 특정 단어 t가 등장한 문장의 수의 역수이다. 각 'article_sent' 변수의 문장별로 등장한 단어를 뽑은 후에 TF-ISF값을 계산하였으며, 계산된 단어별 TF-ISF값의 평균을 구해 그것을 해당 문장의 TF-ISF값으로 하였다. TF-ISF값은 모든 문장에서 자주 등장하는 단어는 중요도가 낮다고 판단하며, 특정 문장에서만 자주 등장하는 단어는 중요도가 높다고 판단하는 중요한 변수이다.[5)]

$$TF\text{-}ISF(w,s) = TF(w,s) * ISF(w)$$

$$Avg\text{-}TF\text{-}ISF(s) = \sum_{i=1}^{W(s)} TF\text{-}ISF(i,s) \,/\, W(s)$$

(3) **sent2sim**: 각 문장들 간의 유사도 값으로, 여기서의 유사도 값은 코사인 유사도(cosine similarity)로 구했다. 이를 구하기 위해선 각 문장의 TF-ISF 행렬이 필요하므로 각 문장을 TF-ISF 행렬로 만든 후 각 기사 내의 문장들 간의 코사인 유사도 값을 구했다.

(4) **ab2sim**: 주어진 데이터에 나와 있던 'abstractive' 변수와 각 문장인 'article_sent' 간의 코사인 유사도 값을 구했다. 마찬가지로 abstractive와 article_sent의 TF-ISF 행렬을 반환한 후 코사인 유사도 값을 구했다.

(5) **SP**(Sentence Position): 각 'article_sent'가 해당 기사 내의 몇 번째 문장인지의 percentile 값이다.

(6) **bias**: 앞서 TextRank 패키지에서 media별로 다른 결과가 나왔음을 이용하여, media별로 train set에서 실제로 요약된 문장들의 빈도를 구한 값이다.

(7) **first**: 각 기사에서 선택된 문장의 분포가 첫 번째가 많아서, 기사의 첫 번째 문장이면 1, 아니면 0을 부여한 변수이다.

---

5) Document Clustering and Text Summarization. Larocca Neto, J.; Santos, A. D.; Kaestner, C.A.; Freitas, A.A.. 2000.

4) 변수 Scaling

- 각 피처의 값이 고르게 퍼져 있어야 더 나은 예측을 할 수 있기 때문에 변수들의 scaling 을 진행하였다. 단 여기서는 모든 변수에 대해 적용하진 않았고, 0 과 1 사이의 값을 갖지 않는 'bias'와 'sent2sim' 2 개의 변수에 대해서만 적용하였다.
- scaling 으로는 표준화, min-max scaler 등을 적용해보았으나 멱변환(Box-Cox Transformation)을 적용하였을 때의 예측 정확도가 가장 높았기 때문에 멱변환을 사용하였다.
- 다음은 변수를 scaling 한 결과이다.

id	first	SL	TF_ISF	ab2sim	SP	transformed_bias	transformed_sent2sim
366196093	1	0.054054	0.057993	0.146774	0.00000	0.360349	-1.073581
366196093	0	0.405405	0.322731	0.160651	0.11111	0.348142	-0.641370
366196093	0	0.540541	0.453334	0.081672	0.22222	0.182758	-0.922482
366196093	0	0.297297	0.299018	0.752785	0.33333	-0.186006	-0.742752
366196093	0	0.648649	0.585356	0.079717	0.44444	-0.437384	-0.944195

5) 모델링 및 예측 결과

- Train set 을 이용해 모델을 학습시킨 후, Validation set 에 대해 성능 평가를 하였다.(공모전이 종료된 이후에 모델링을 하였기 때문에 Test set 에 대한 성능 평가를 할 수는 없었다.)
- 평가 기준은 정확도(Accuracy)이며, 더 나은 성능을 위해 GridSearch 를 이용해 하이퍼 파라미터 튜닝하였다.
- 결과는 다음과 같다.

Decision tree classifier	0.893
Random Forest	0.90
Naive Bayes	0.87
LightGBM with Bayesian Optimization	**0.91**
앞의 4 개 모델에 대한 Voting Classifier	**0.91**

- 결과적으로 **LightGBM 과 Voting Classifier** 의 성능이 91%로 가장 좋았다.