zero-base

문장 유형 분류 ML

이석호 , 이희웅 , 하승범 , 조예람

Contents

01 주제

문장 유형 분류

: 자연어처리

다양한 머신러닝 모델을 이용한 문장 분석

02 데이터

문장분류 데이터 소개

- 출처: **Dacon 문장분류 AI 경진대회 제공 데이터**
- 레이블 세부 분류

유형	극성	시제	확실성
- 사실형 - 추론형 - 대화형 - 예측형	- 긍정 - 부정 - 미정	- 과거 - 현재 - 미래	- 확실성 - 불확실성

예시:

정부가 고유가 대응을 위해 7월부터 연말까지 유류세 인하 폭을 30%에서 37%까지 확대한다.	사실형	긍정	미래	확실	사실형-긍정-미래-확실
서울시는 올해 3월 즉시 견인 유예시간 60분을 제공하겠다고 밝혔지만, 하루 만에 차도와 자전거도로는 예외로 하겠다고 입장을 바꾸기도 했다.	사실형	긍정	과거	확실	사실형-긍정-과거-확실

03 EDA

사실형: 13558
추론형: 2151
예측형: 257
대화형: 575

사실형

추론형

예측형

대화형

- **사실형: ~있다, ~했다, ~이다, ~인다,**
- **추론형: ~야 한다, ~으로 봤다, ~수 있다, ~것이다, ~때문이다, ~지도 모른다**
- **예측형: 전망이다, 있겠다, ~보인다, ~하겠다, ~예정이다, 전망한다, 전망된다, 기대된다**
- **대화형: ~하라, ~입니다, ~습니다, ~바랍니다, 있었죠, 되나, 말이죠, ~일까?**

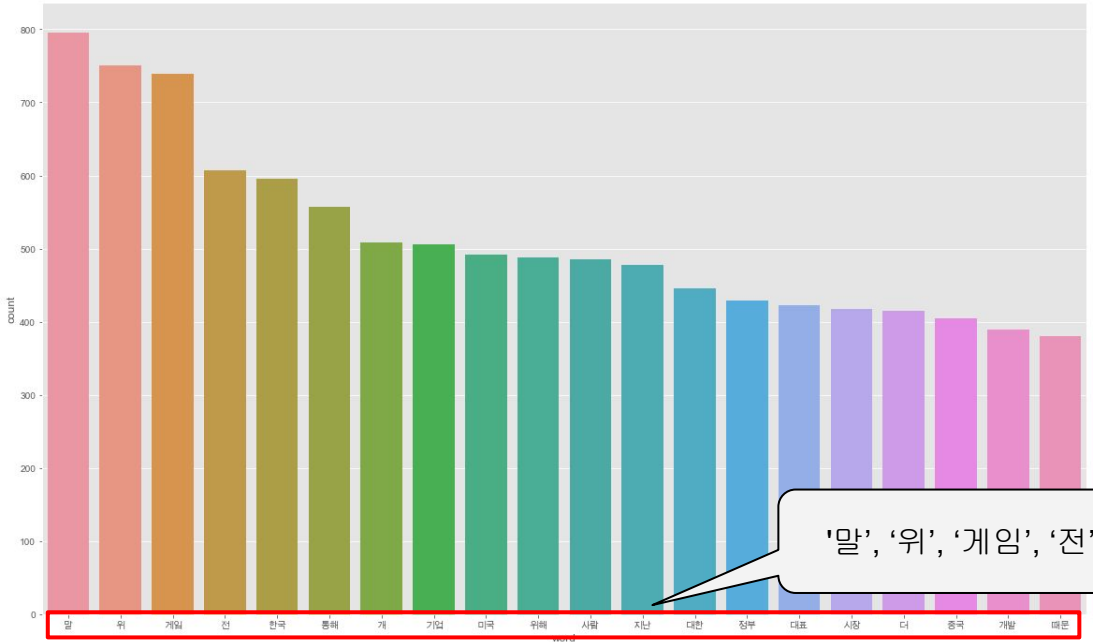
- **대화형에서는 문장부호 (?, !) 자주 등장**

- **분류할 때 문장의 명사나 고유명사보다 문장 전체의 구조에 더 집중**

  예시: _____에 가면 보통 _____하니까 _____할 것이다.

  - 어떤 단어가 들어와도 예측형이라는 사실은 변하지 않음

주어진 데이터 살펴보기



'말', '위', '게임', '전', '한국', '통해', '개', '기업', '미국', '위해'…

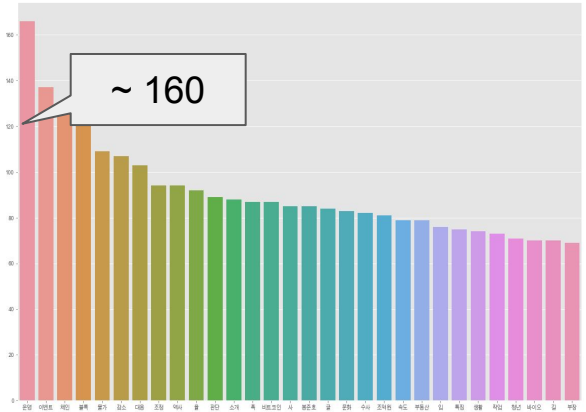
단어 구분 후 가장 많이 등장한 단어 **20**개

훈련 데이터와 단어(명사) 수

훈련 데이터 : **16,541**

단어(명사) 수 : **20,570**

각 극성에 따른 단어 수 살펴보기

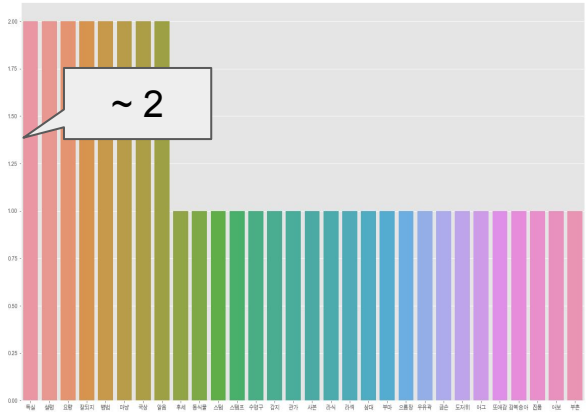


긍정

긍정 문장 : 15,793

긍정 문장 단어 : 20,418

(운영, 이벤트, 체인, 블록,
물가, 감소…)

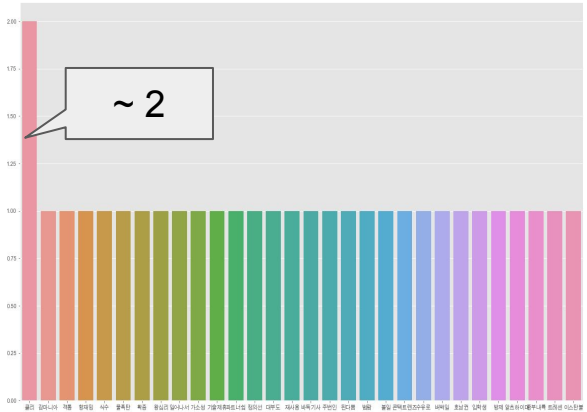


부정

부정 문장 : 565

부정 문장 단어 : 2,744

(득실, 설령, 요량, 잘되지,
마냥, 국상…)



미정

미정 문장 : 183

미정 문장 단어 : 1,252

(격통, 식수, 확증, 범람,
볼일…)

극성 기준에서의 EDA

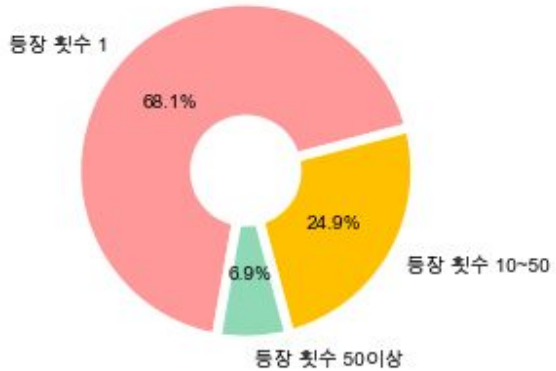
전체 데이터를 단어로 분리하여 불용어를 제거하더라도 훈련 데이터의 문장에는 30% 이상의 단어들이 10회 이상 중복하여 등장하고 있다. 가장 많이 등장 한 횟수는 796번이다.

전체 데이터로 보면 데이터 자체는 충분해 보이지만 극성(긍정, 부정 미정)으로 구분지어 데이터를 확인해보면 긍정 카테고리에 95% 데이터들이 속한다.

훈련 데이터 문장을 단어로 분리하여 보면 대부분의 단어들이 긍정으로 분류된 문장안에 등장 하는 것을 알 수 있었다.

부정, 미정으로 분류된 그룹은 전체적인 데이터 수와 단어들의 빈도수에서 긍정 데이터와 불균형이 있다.

따라서 모델 형성을 위한 전처리에서 데이터 불균형은 필수로 고려해야할 사항이다.

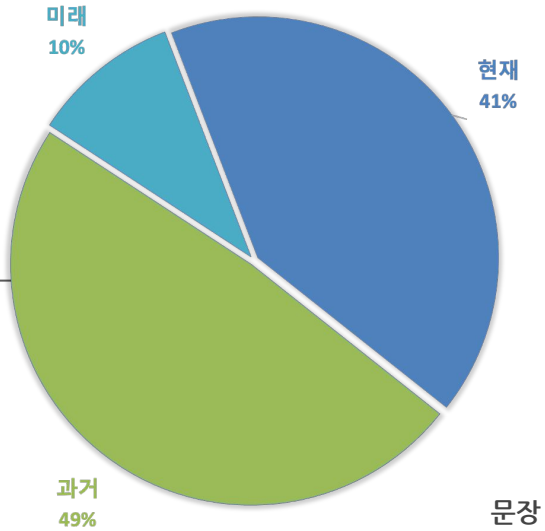


단어 빈도수

총 단어 수 : 20,570 개

최대 / 최소	796회 / 1회
평균	약 10회

03. EDA - 시제



현재: 6850 과거: 8014 미래: 1642

문장 별 시제를 구분 가능하게 하는 구간

현재: 경기 악화로 인해 P2P 업체 대출의 부실화 가능성이 커지고 있는 것도 위험 요인으로 꼽히고 있다.
과거: 퇴계 이황은 학문에서도 따라올 자가 없었지만 인품도 뛰어나 당파를 가리지 않고 수많은 학자들의 존경을 받았다.
미래: "나는 천지를 관(棺)으로 삼고, 해와 달을 벗으로 삼으며 별을 보석으로 삼고, 만물을 부장품으로 삼을 것이다."

현재: 또 원작 '히트' 이후의 세계관을 담은 MMORPG '히트2'의 경우 원작보다 더욱 방대해진 세계관 속에서 이용자들이 장대한 여정에 오르는 느낌을 받을 수 있도록 구성한 음악의 탄생 배경과 작곡 과정을 설명할 예정이다.
과거: 자연 속성의 5성 전사로 아군 생존력 강화 능력을 갖췄다.
미래: 향후 필요 시 추가적인 자본확충을 지속적으로 한다는 계획이다.

현재: 권분을 강제하는 것은 위험하다.
과거: 산업부는 최근 신종 코로나바이러스 사태를 예의주시하며 내부 검토를 진행해온 것으로 알려졌다.
미래: 예상 강수량은 5~40㎜로, 충남서해안과 전라권서부엔 5㎜ 내외의 비가 내리겠다.

- 과거 : ~았지만, ~했지만, ~렸다. ~했다. 않았다. 설명했다. ~이었다.

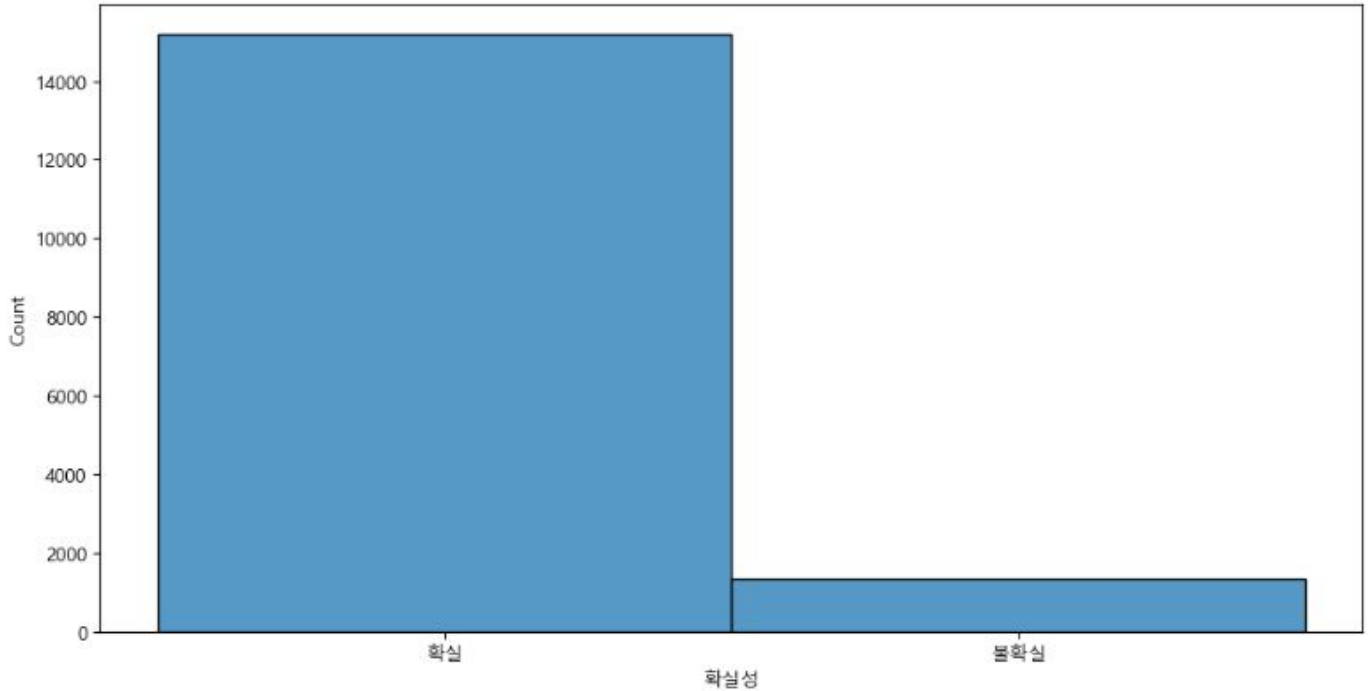
- 현재 : ~진다. ~한다. 없다. 놀랍다. ~중입니다.

- 미래 : ~예정이다. 예상된다. 계획이다. 될 수 있다.

=> 마지막 종결어미가 시제에 큰 영향을 미친다 판단.

“확실” 데이터와 “불확실” 데이터 수의 차이가 크게 나타난다.

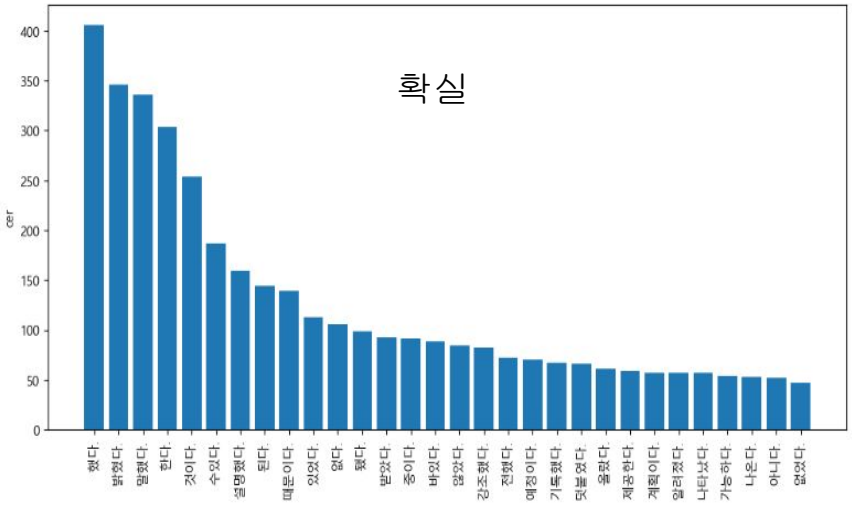
(확실: 15192, 불확실: 1349개)

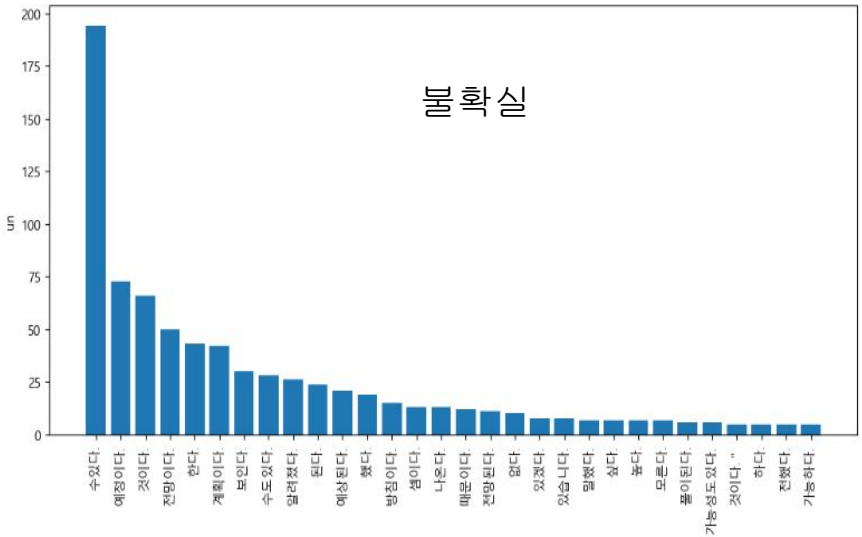


문장의 확실성을 분류하기에 단어의 의미는 크게 영향을 미치는 것 같지 않다.

- 단어 단위 보다는 문장의 서술어 부분이 중요.
- 확실성 문장: ~이다. ~있다. ~했다.
- 불확실성 문장: ~계획이다. ~예정이다. ~수 있다. ~것이다. ~수도 있다.

문장의 확실성을 나누는 기준이 **문장의 추측성(양태)와 밀접한 관련**이 있다고 생각된다.





04 전처리

문장 유형 전처리

예시 문장:

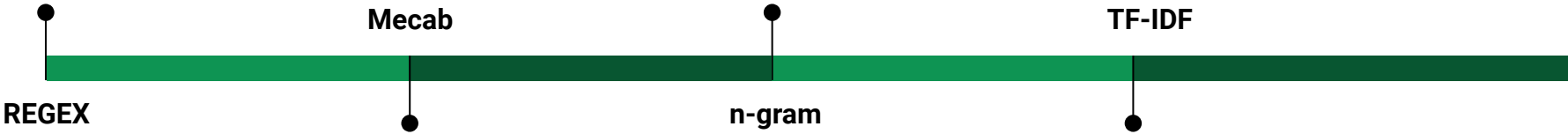
"한국게임미디어협회(KGMA) 신년기획 10부작 게임질병 코드 어떻게 볼 것인가 1부, 왜 게임은 탄압받는가?"

REGEX로 한글, !, ?만 필터링

한국게임미디어협회 신년기획 부작
게임질병 코드 어떻게 볼 것인가 부 왜
게임은 탄압받는가?

n-gram화 (1 ~ 3)

['한국', …
'게임 미디어', … ,
'받는가?']



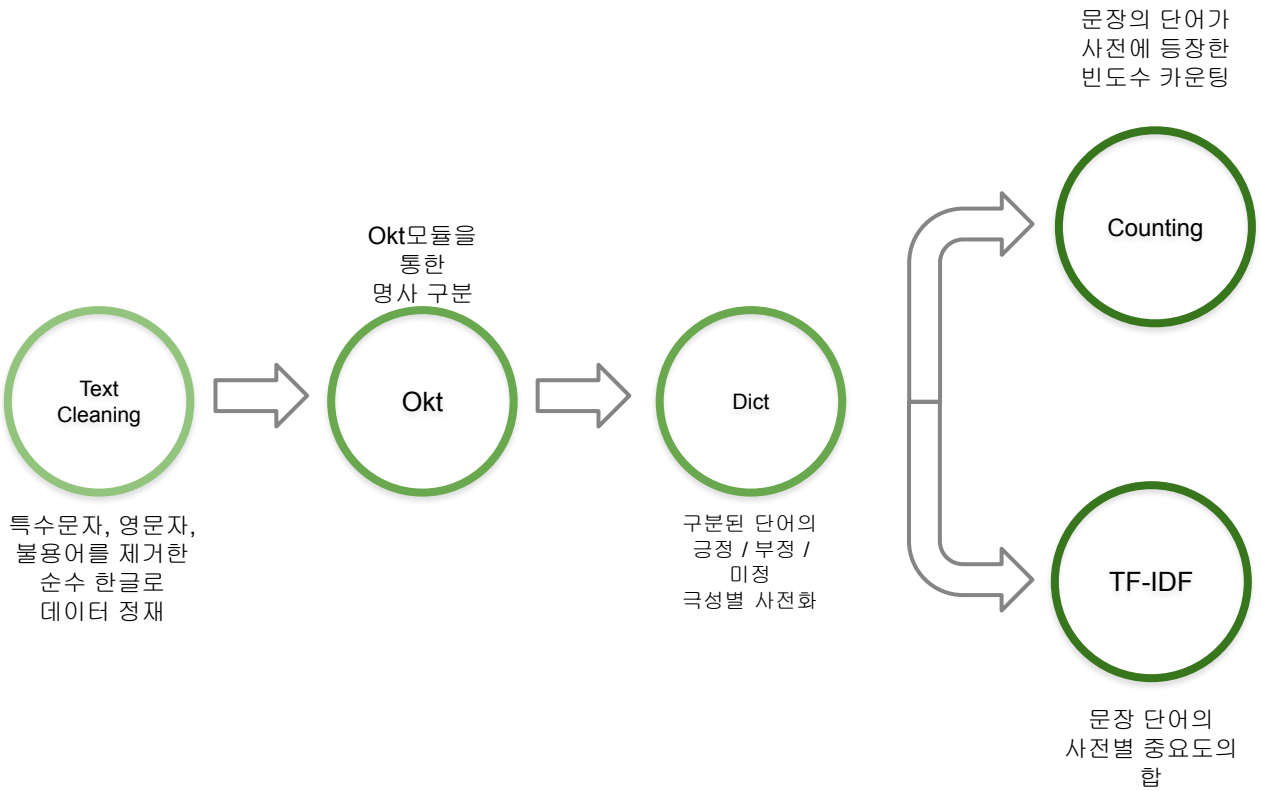
문장 토큰화

['한국', '게임', '미디어', '협회',
…, '받', '는가', '?']

토큰 중요도 설정

[0.0, 0.0, 0.0589, 0.021, 0.0,
0.0 …. 0.0, 0.0, 0.0]

극성에 따른 데이터 전처리



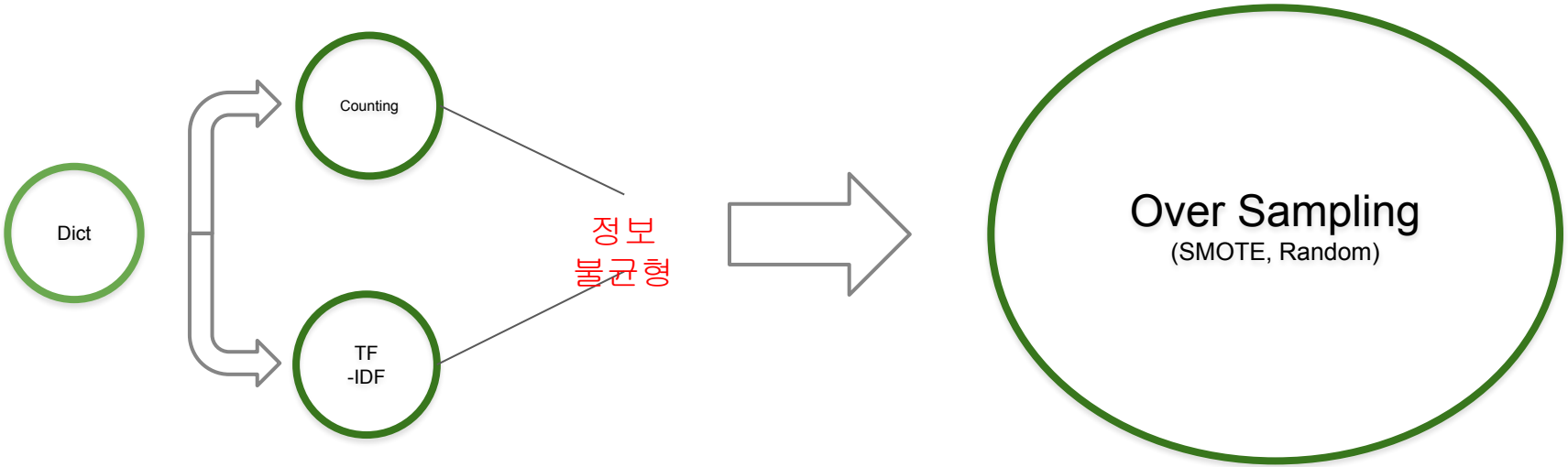
	pos	nega	neu	True
0	5	4	5	긍정
1	22	15	10	긍정
2	10	7	3	긍정
3	14	11	6	긍정
4	6	4	0	긍정
...	...	...	...	...
16536	16	4	5	긍정
16537	21	11	5	긍정
16538	13	7	2	긍정
16539	18	12	8	긍정
16540	2	1	0	긍정

16541 rows × 4 columns

	pos	nega	neu
0	0.244787	0.254119	0.096107
1	0.000000	0.000000	0.000000
2	0.273984	0.220236	0.109837
3	0.201252	0.152471	0.123566
4	0.005214	0.016941	0.000000
...	...	...	...
16536	0.000000	0.000000	0.000000
16537	0.015120	0.042353	0.054918
16538	0.135558	0.076236	0.054918
16539	0.158499	0.101647	0.054918
16540	0.042753	0.008471	0.000000

16541 rows × 3 columns

극성에 따른 데이터 전처리



04. 전처리 - 시제

중복 데이터 제거

- train데이터 내에 같은 문장이지만 시제 명 다른 데이터 발견.
- 총 35*2 => 70개 중, 중복 문장 삭제

Text Cleansing

- 단순 자음과 모음 등, 한글을 제외한 문자 제거.

Customize Tokenizer

- Konlpy의 Okt, HANNANUM 사용.
- 시제 판별에 가장 큰 영향을 준다고 생각되는 형태소만 뽑아 새로운 칼럼에 추가
-

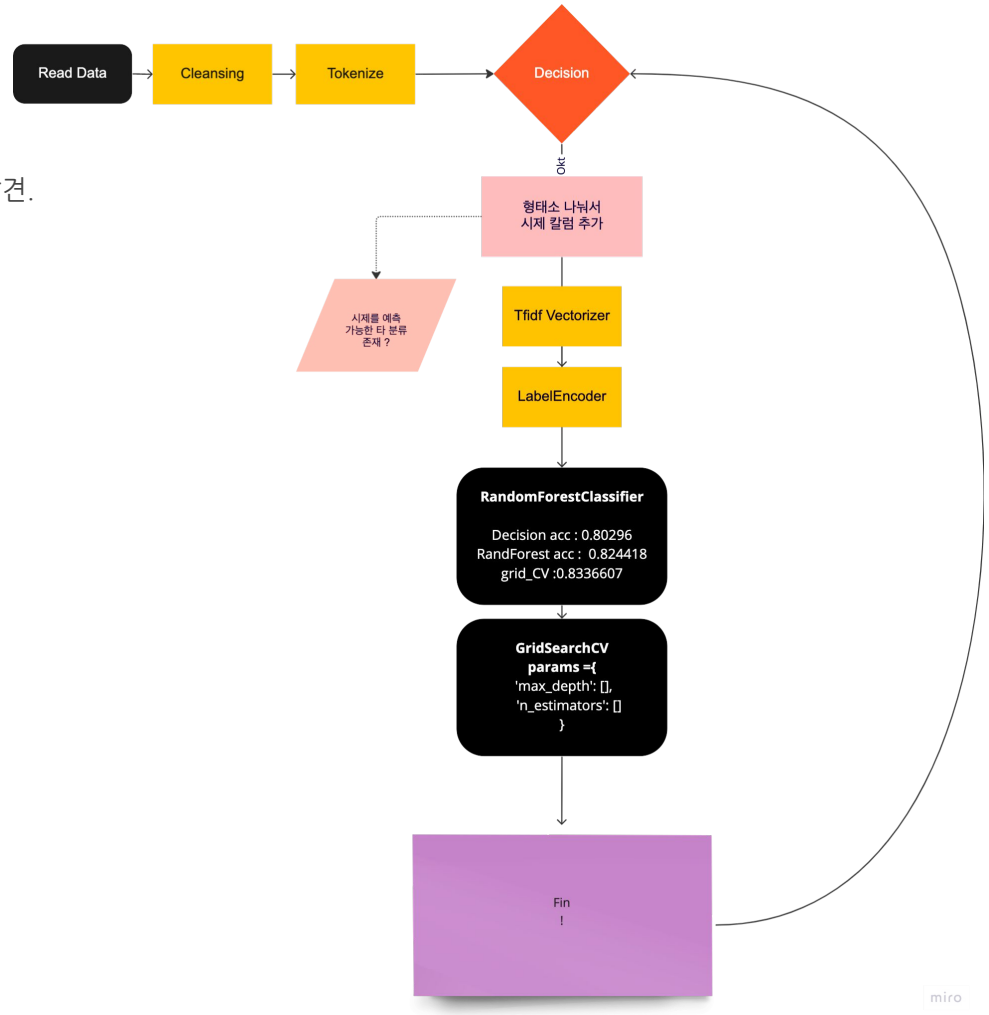
```
3    '하겠다고', '밝혔지만', '만에', '하겠다고', '바꾸기도', '했다'
4          '자는', '태워', '놓고', '으로', '막아', '채운다'
5             '같은', '에서', '으로', '줄어든다는', '이다'
```

TfidfVectorizer

- 위에서 생성한 칼럼의 value값만 뽑아 vectorize 진행

LabelEncoder

- train['시제']



04. 전처리 - 확실성

1. 문장의 끝 2어절 추출

예시)

"말씀 중 죄송합니다만 우리 흥민이 절대 월드클래스 아닙니다."

"월드클래스 아닙니다 "

	문장 전체	문장 끝 2어절
f1_score	0.8421665695981362	0.8566153846153847
roc_auc_score	0.6208240644521449	0.6992872016342206

2. Tf-idf 벡터화

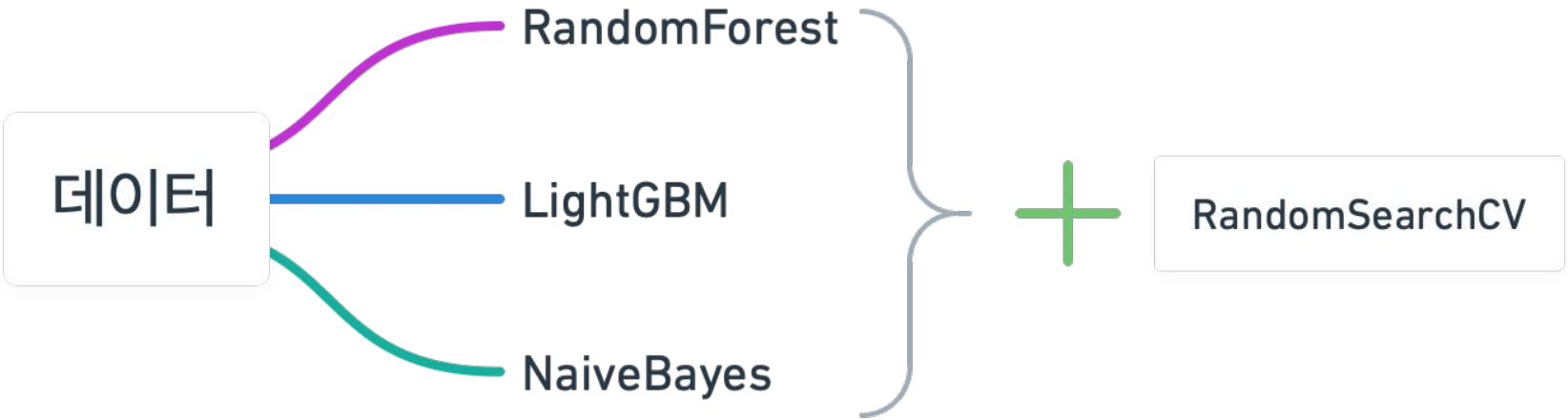
1번 과정에서 추출한 문장을 Tf-idf 벡터화
=> X 데이터

3. Label Encoder

“확실”을 1, “불확실”을 0으로하는 새로운 y 컬럼 생성
=> y 데이터

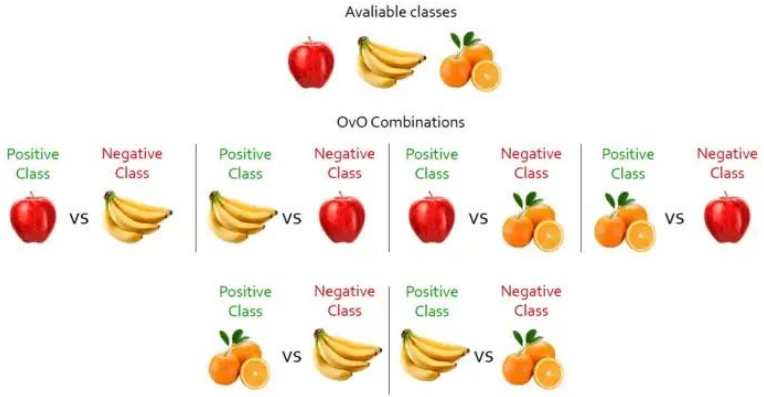
05 모델

유형분석 - 모델 실험

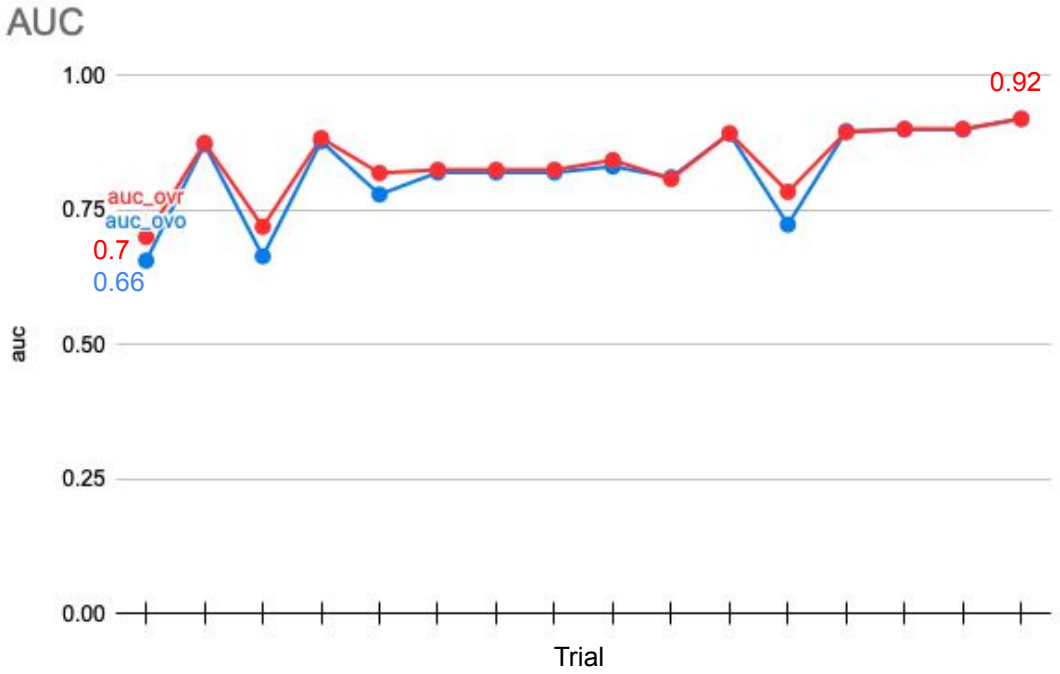


유형분석 - 평가지표

OvO / OvR AUC

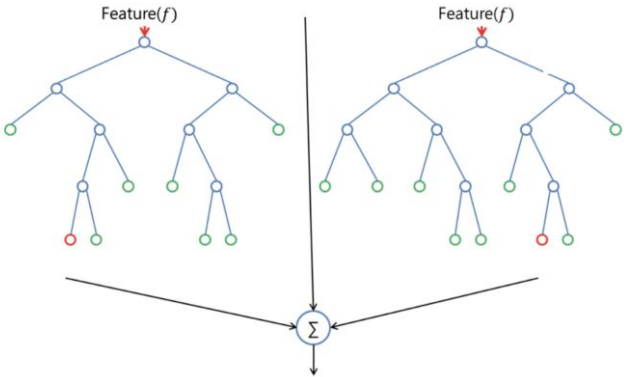


출저: Vinícius Trevisa, Multiclass classification evaluation with ROC Curves and ROC AUC



최종 모델

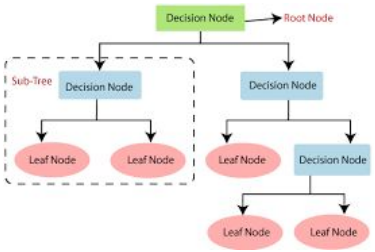
Random Forest



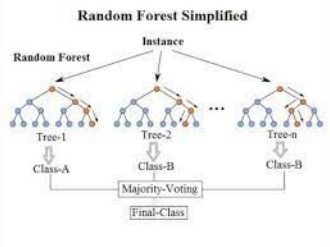
- 파라미터
  - Gini
  - min_sample_split:10
  - n_estimators=399
  - max_depth=None

사용 모델

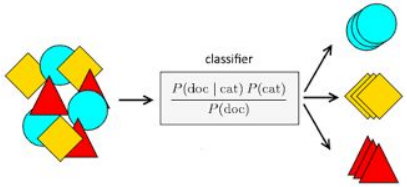
Decision Tree



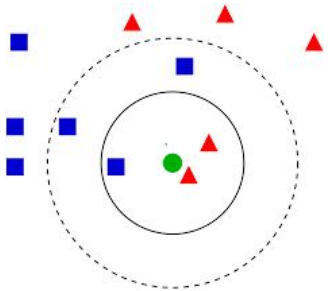
RandomForest



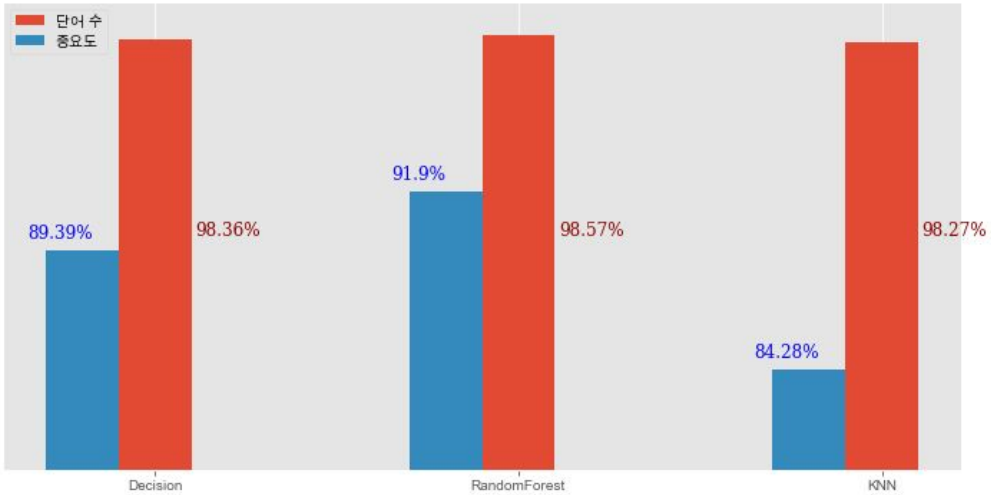
Naive bayes



KNN



사용 모델 - 정확성 비교



모델	정확성	
	단어수	중요도
Decision	98.36%	89.39%
RandomForest	**98.57%**	**91.90%**
KNN	98.27%	84.28%

+

Naive bayes : 87.39%

사용 모델 - 극성별 정확도(부정, 미정)

단어 수

모델	정확성		
	전체	부정	미정
Decision	98.26%	**99.11%**	96.17%
RandomForest	98.57%	98.76%	97.26%
KNN	**98.27%**	98.23%	**98.90%**

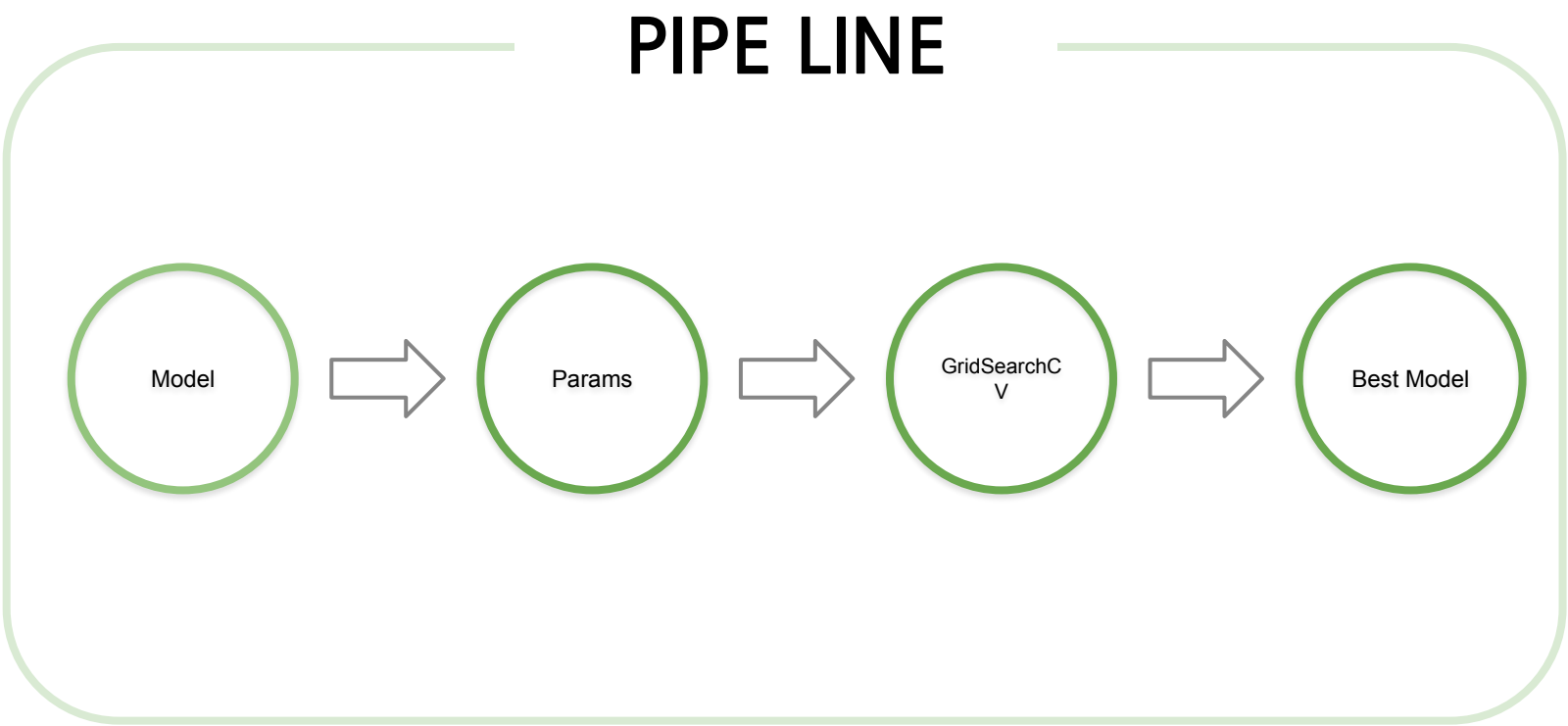
중요도 합

모델	정확성		
	전체	부정	미정
Decision	89.39%	91.32%	91.25%
RandomForest	**90.90%**	93.27%	**92.89%**
KNN	84.25%	**95.04%**	**92.89%**

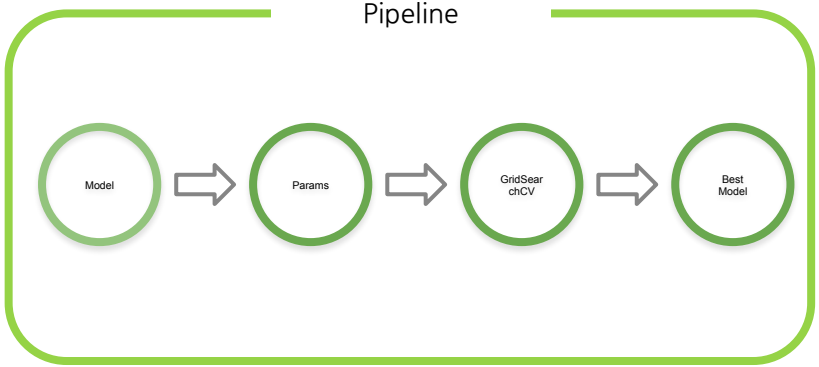
+

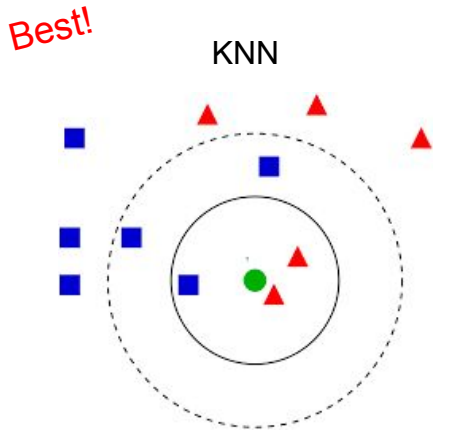
Naive bayes - 부정 : 92.21% 미정 : 89.61%

사용 모델 - 하이퍼 파라미터 찾기



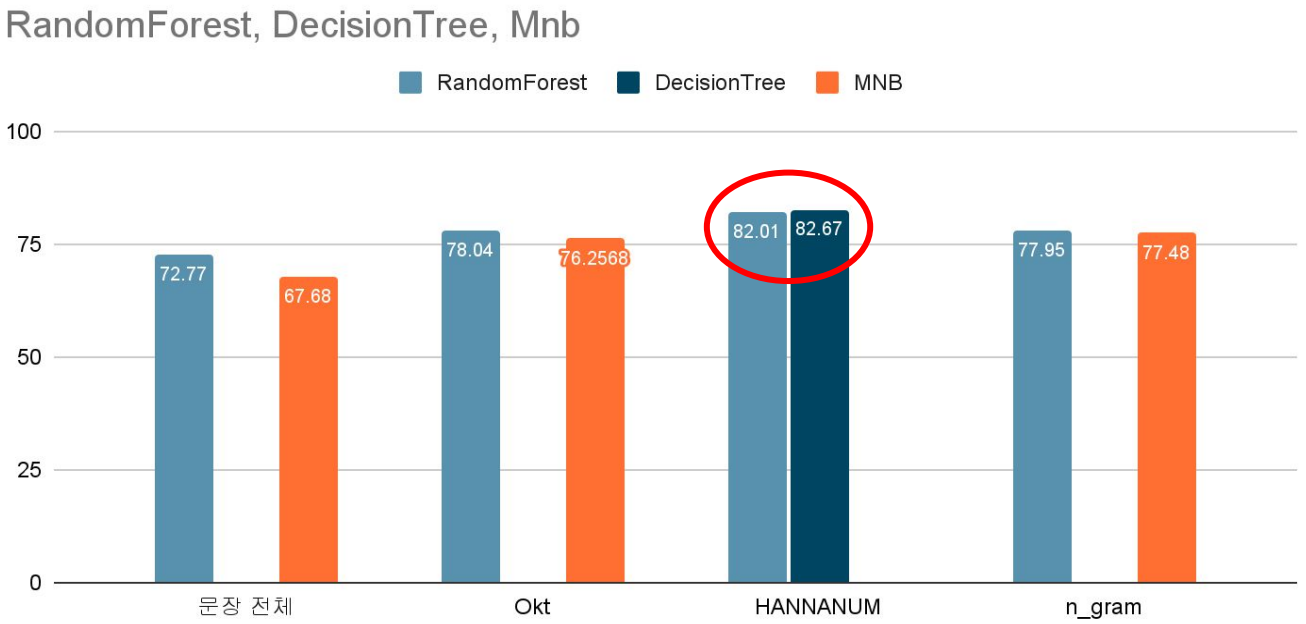
사용 모델 - 하이퍼 파라미터 찾기





n_neighbors : 15, weights: distance

05. 모델 - 시제


RandomForest, DecisionTree, Mnb
RandomForest
DecisionTree
MNB
100
75
50
25
0
72.77
67.68
78.04
76.2568
82.01
82.67
77.95
77.48
문장 전체
Okt
HANNANUM
n_gram

05. 모델 - 시제

tfidf												
Min_df	5	10	0.2	15	15	10	10	10	10	1	10	10
n_gram_range	(1,3)	(1,4)	(1,2)	(1,5)	(1,5)	(1,4)	(1,4)	(1,4)	(1,4)	(1,2)	(1,5)	(1,4)
max_df			0.8	0.8								
max_features			10000	10000								
analyzer = 'word'					word	word	word					
grid_cv							RandomForestClassifier					
max_depth	50	50		50	50	50	90		50		257	837
n_estimators	130	130		130	130	150	114		130		393	88
							criterion': 'entropy' 'min_samples_leaf': 2				'entropy' 'min_samples_leaf': 2 'min_samples_split': 11	'gini' 'min_samples_leaf': 3 min_samples_
Sco	0.83542	0.8356		0.8336	0.834292	0.835656	0.83815				0.8380796	0.8380796
RandomForest Sco	0.8201	0.81768	0.7764	0.816172	0.8146	0.81798		0.6486	0.745	0.75942	0.8198	0.817
MNB sco	0.7292	0.73107	0.529678	0.728	0.72804	0.7310721		0.638111		0.65275	0.731	
DecisionTree Sco	0.82586	0.82646	0.77649	0.826165	0.8261	0.826468		0.7483		0.65056	0.8264	0.826468
pred X_val					0.814657	0.81798					0.770139	

로지스틱 회귀

‘C’ : [0.01, 0.1, 1, 5, 10]
‘max_iter’ : [200, 500, 1000]

StratifiedKFold(n_splits=5)

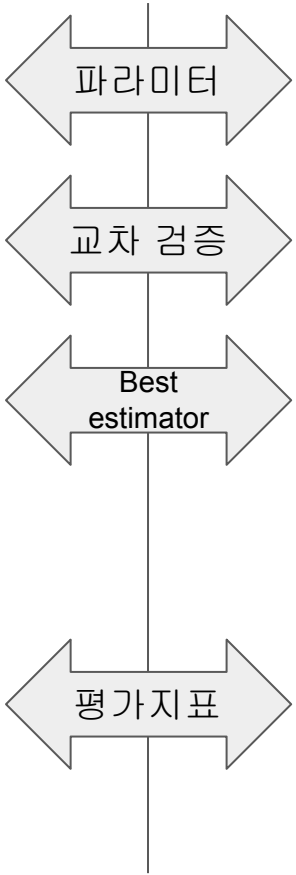
'C': 5
'max_iter': 200

accuracy_score_train : 0.9395890831033425
accuracy_score_test : 0.7858455882352942

f1_score_train : 0.9565025391918747
f1_score_test : 0.8566153846153847

recall_score_train : 0.9630947087594487
recall_score_test : 0.9267643142476698

roc_auc_score_train : 0.9252232437077876
roc_auc_score_test : 0.6992872016342206



나이브베이즈 - MultinomialNB

'alpha': [0.01, 0.1, 1, 5, 10]

StratifiedKFold(n_splits=5)

‘alpha’ : 1

accuracy_score_train : 0.8972707758356332
accuracy_score_test : 0.7784926470588235

f1_score_train : 0.9286170892819092
f1_score_test : 0.8522378908645004

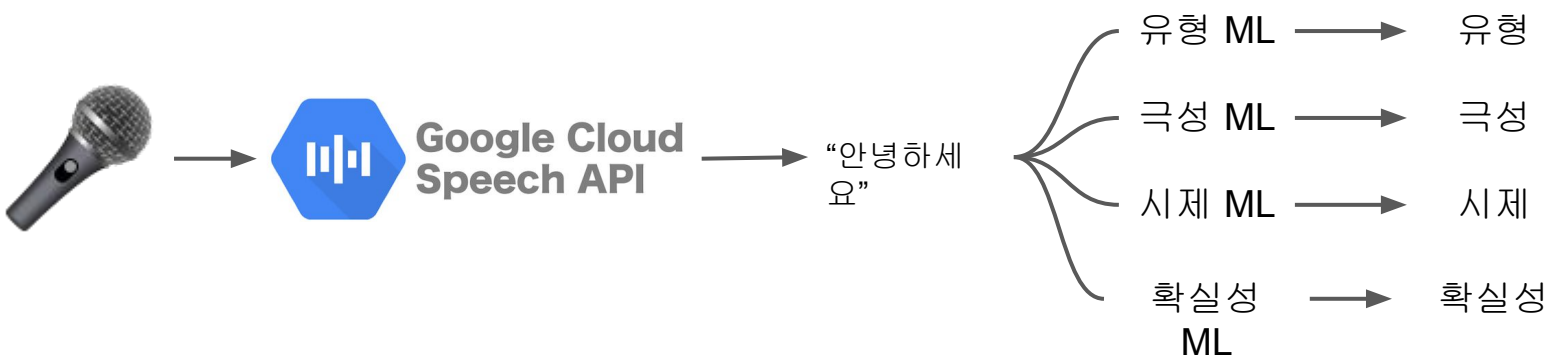
recall_score_train : 0.9688750555802579
recall_score_test : 0.9254327563249002

roc_auc_score_train : 0.8535086740351884
roc_auc_score_test : 0.6882356659962781

06 결과

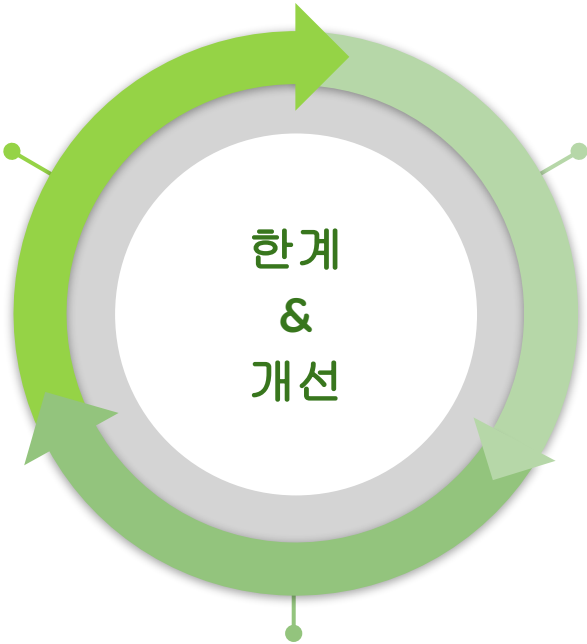
전체 모델 적용

- Google Cloud STT(Speech-To-Text) API 적용
- 사용자 화법의 유형, 극성, 시제, 확실성을 분석



07 결론

한계점 및 개선 사항



데이터 최신화
신조어 등 새 데이터로 모델 최신화

머신러닝의 한계
자연어의 복잡성

데이터 다양성
구어체, 문어체 등 다양한 유형의 문장 데이터 필요

감사합니다