패턴인식 프로젝트 보고서

9 팀 2122004 고은서 2176225 원재영 2271107 이경선 2271104 최수희

목차

---

2. 파생 변수 생성
    2-1. 파생 변수 생성 함수 정의
    2-2. 파생 변수 적용
3. 데이터셋 분할 (훈련/검증)

---

4. 모델링 및 학습

1. 최적 임계값
2. 사용한 모델 개요
    2-1. Base learners (LightGBM, XGBoost, CatBoost)
    2-2. Stacking 모델 구조
    2-3. 메타 모델(XGBoost)
3. 클래스 불균형 처리 방법 - SMOTE 오버샘플링
4. 모델 학습 과정

---

5. 결과 및 성능 평가

1. 예측 결과
2. 최적 임계값 설정
3. 결과 해석
4. 테스트 세트 예측 클래스 분포

---

6. 결론 및 시사점

1. 개요

---

1. 프로젝트 목표

본 프로젝트의 주요 목표는 뉴스 콘텐츠 관련 데이터를 기반으로 해당 콘텐츠가 바이럴(popular)될 가능성이 있는지를 예측하는 이진 분류(Binary Classification) 모델을 개발하는 것을 목표로 한다. 기사의 인기도는 shares 변수(기사 공유 수)를 기준으로 정의되며 1400 회 이상 공유된 경우를 '인기 있음(1)' 로 간주하고, 그 외는 '인기 없음(0)'으로 처리하여 y 라는 목표 변수로 구성하였다.

---

2. 데이터 설명 및 문제 정의

데이터는 총 22,200 건의 학습 데이터(train.csv)와 9,515 건의 테스트 데이터(test.csv)로 구성되어 있으며, 각 각은 다음과 같은 구조를 가진다.

- train.csv
    - 총 49 개의 컬럼
    - 1 개의 식별자 컬럼 + 46 개 입력 피처 + 2 개 출력 관련 변수 (y, shares)
- test.csv
    - 총 47 개의 컬럼
    - 1 개의 식별자 컬럼 + 46 개 입력 피처

입력 변수는 기사 내용의 길이, 이미지 및 비디오 수, 키워드 통계, 단어 사용 특성 등 다양한 콘텐츠 기반 특성으로 구성되어 있다.

목표 변수인 y 는 shares 에서 파생된 이진 값으로, 실제 shares 변수는 참고용으로만 제공되며 학습에 사용되지 않는다. 또한, 입력 변수는 약 10%정도의 결측값을 포함하고 있어, 적절한 결측치 처리 전략이 필요하다. 이러한 점을 고려할 때 본 문제는 다차원 수치와 범주형 혼합 피처와 결측치, 클래스 불균형 문제까지 포함된 복합적인 분류 문제라고 할 수 있다.

2. 탐색적 데이터 분석 (EDA)

---

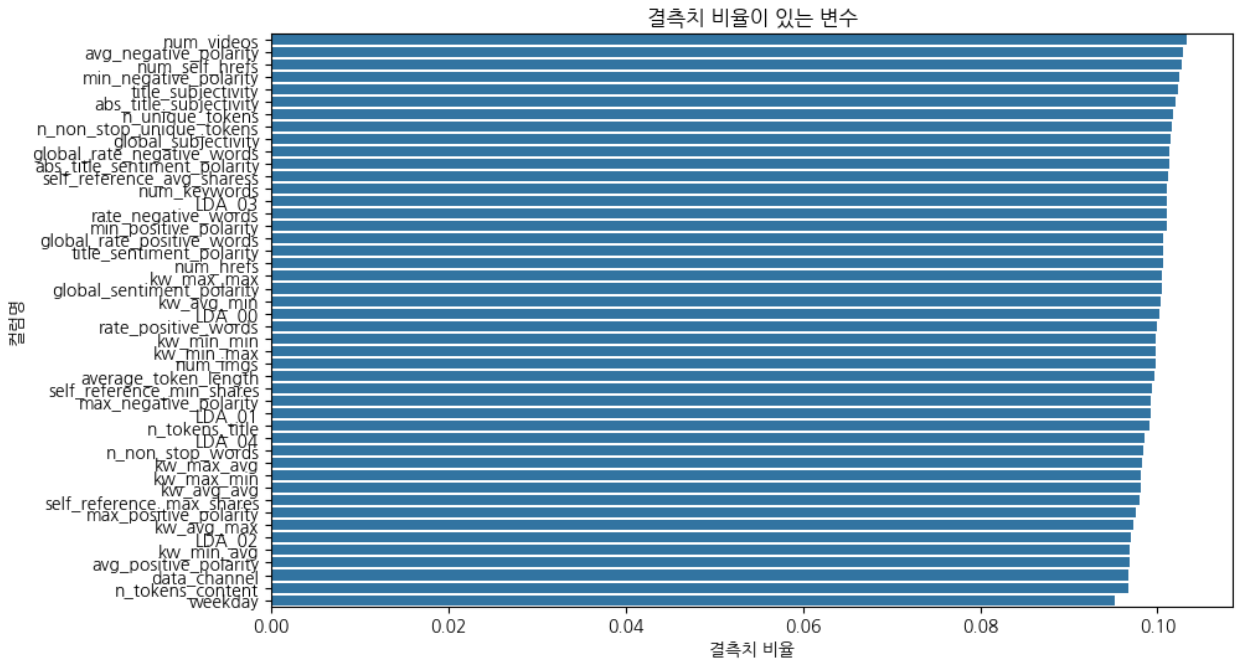
1. 데이터 기초 정보

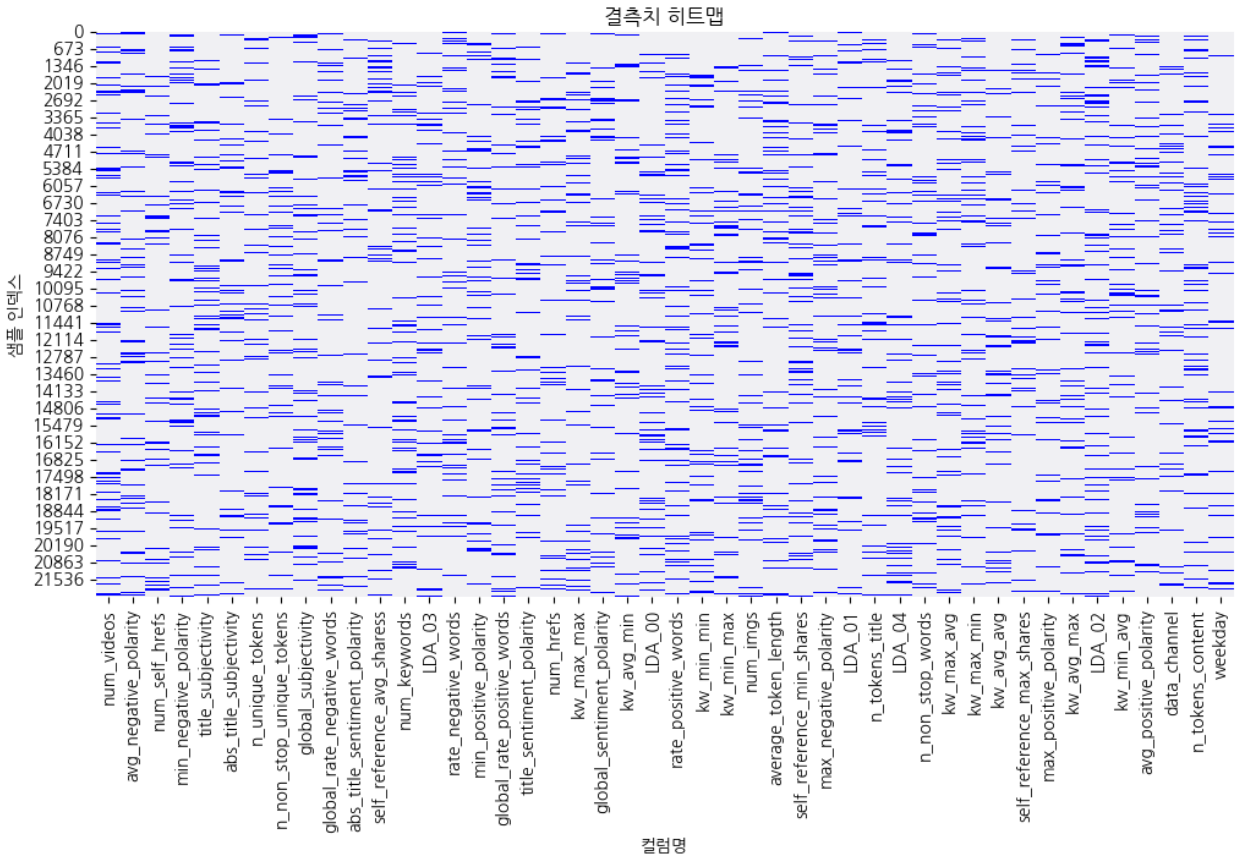
데이터는 총 22,200 건, 47 개 변수(shares, y 제외)로 이루어져 있고, 수치형, 범주형 변수를 모두 포함한다. 타겟 변수 y 는 이진 분류 (0 = 비인기, 1 = 인기)이다.

---

2. 결측치 분석

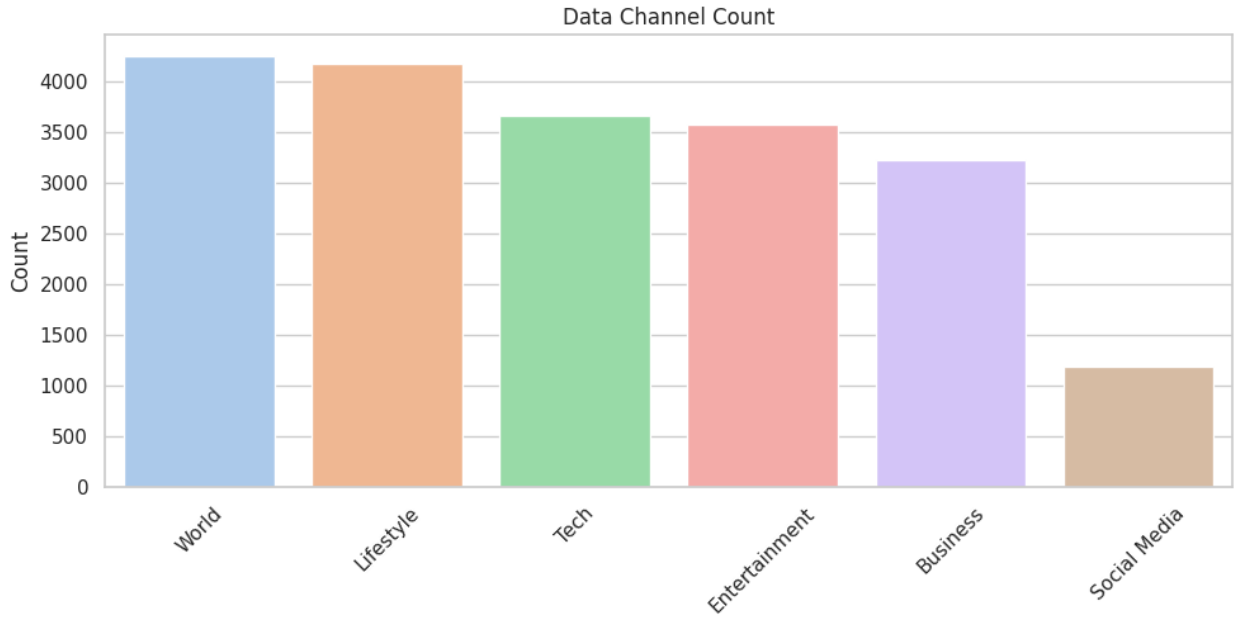
47 개 모든 변수에 약 10% 결측률의 결측치가 있다. 수치형 변수는 중앙값 또는 평균으로 대체하고, 범주형 변수는 최빈값 또는 'unknown'으로 대체할 필요가 있다. 결측치가 고르게 분포하여 샘플을 삭제할 필요는 없다.

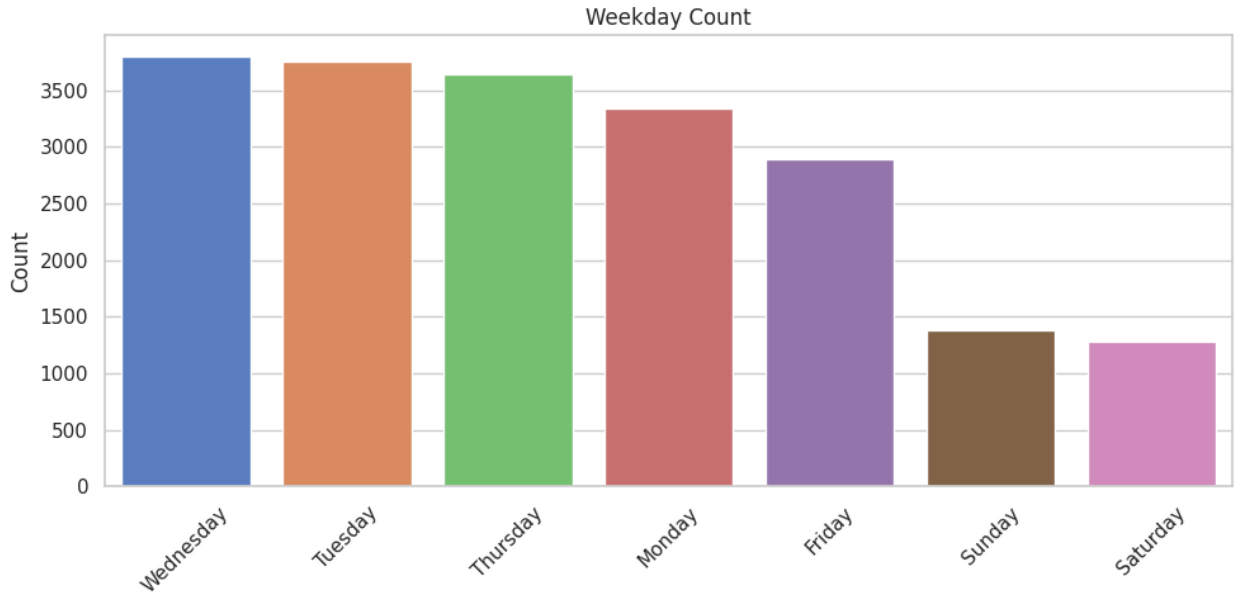



결측치 히트맵
샘플 인덱스
0
673
1346
2019
2692
3365
4038
4711
5384
6057
6730
7403
8076
8749
9422
10095
10768
11441
12114
12787
13460
14133
14806
15479
16152
16825
17498
18171
18844
19517
20190
20863
21536
num_videos
avg_negative_polarity
num_self_hrefs
min_negative_polarity
title_subjectivity
abs_title_subjectivity
n_unique_tokens
n_non_stop_unique_tokens
global_subjectivity
global_rate_negative_words
abs_title_sentiment_polarity
self_reference_avg_sharess
num_keywords
LDA_03
rate_negative_words
min_positive_polarity
global_rate_positive_words
title_sentiment_polarity
num_hrefs
kw_max_max
global_sentiment_polarity
kw_avg_min
LDA_00
rate_positive_words
kw_min_min
kw_min_max
num_imgs
average_token_length
self_reference_min_shares
max_negative_polarity
LDA_01
n_tokens_title
LDA_04
n_non_stop_words
kw_max_avg
kw_max_min
kw_avg_avg
self_reference_max_shares
max_positive_polarity
kw_avg_max
LDA_02
kw_min_avg
avg_positive_polarity
data_channel
n_tokens_content
weekday
컬럼명

3. 범주형 변수 분석

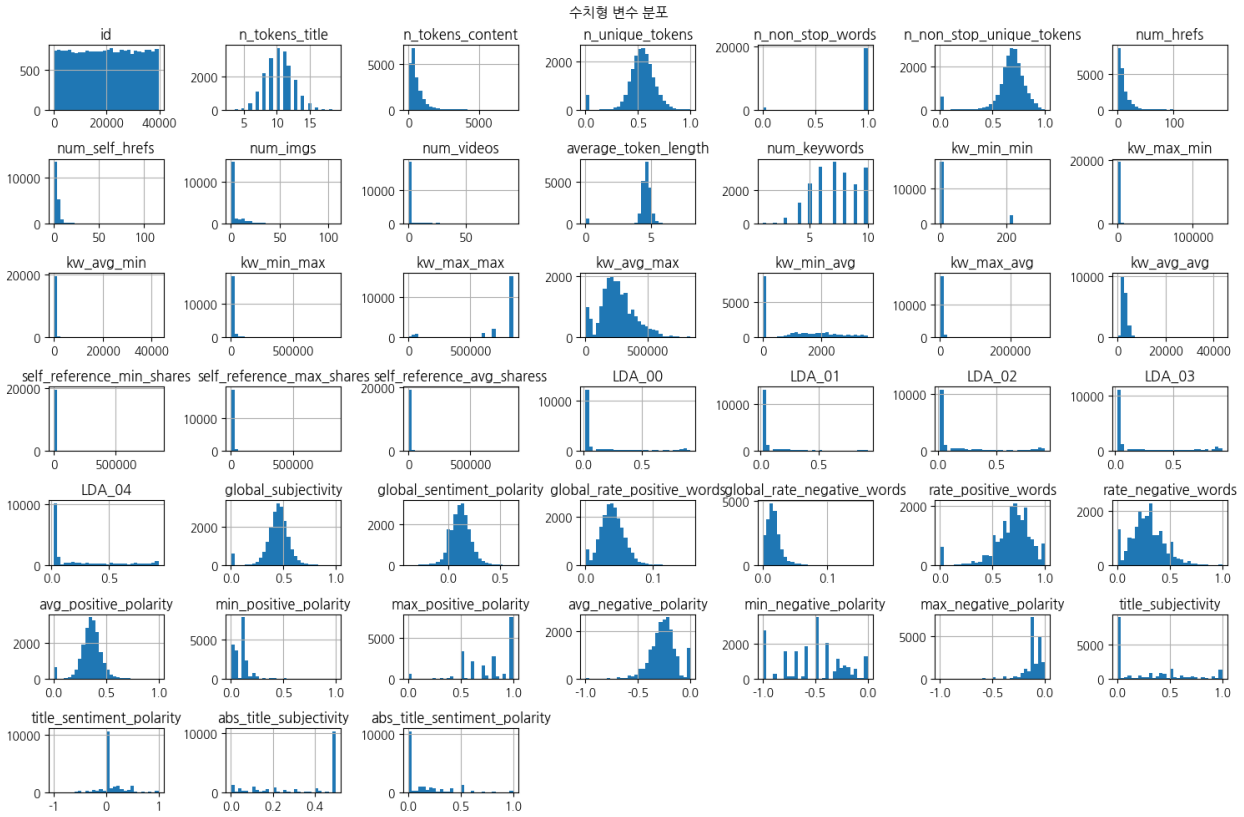
범주형 변수는 data_channel, weekday 이다. data_channel 은 6 개 클래스 (World, Lifestyle, Tech, Entertainment 등)로 이루어져 있고, weekday 는 요일 (Monday~Sunday)이다. 이후 One-hot encoding 또는 Label Encoding 으로 전처리할 수 있다.





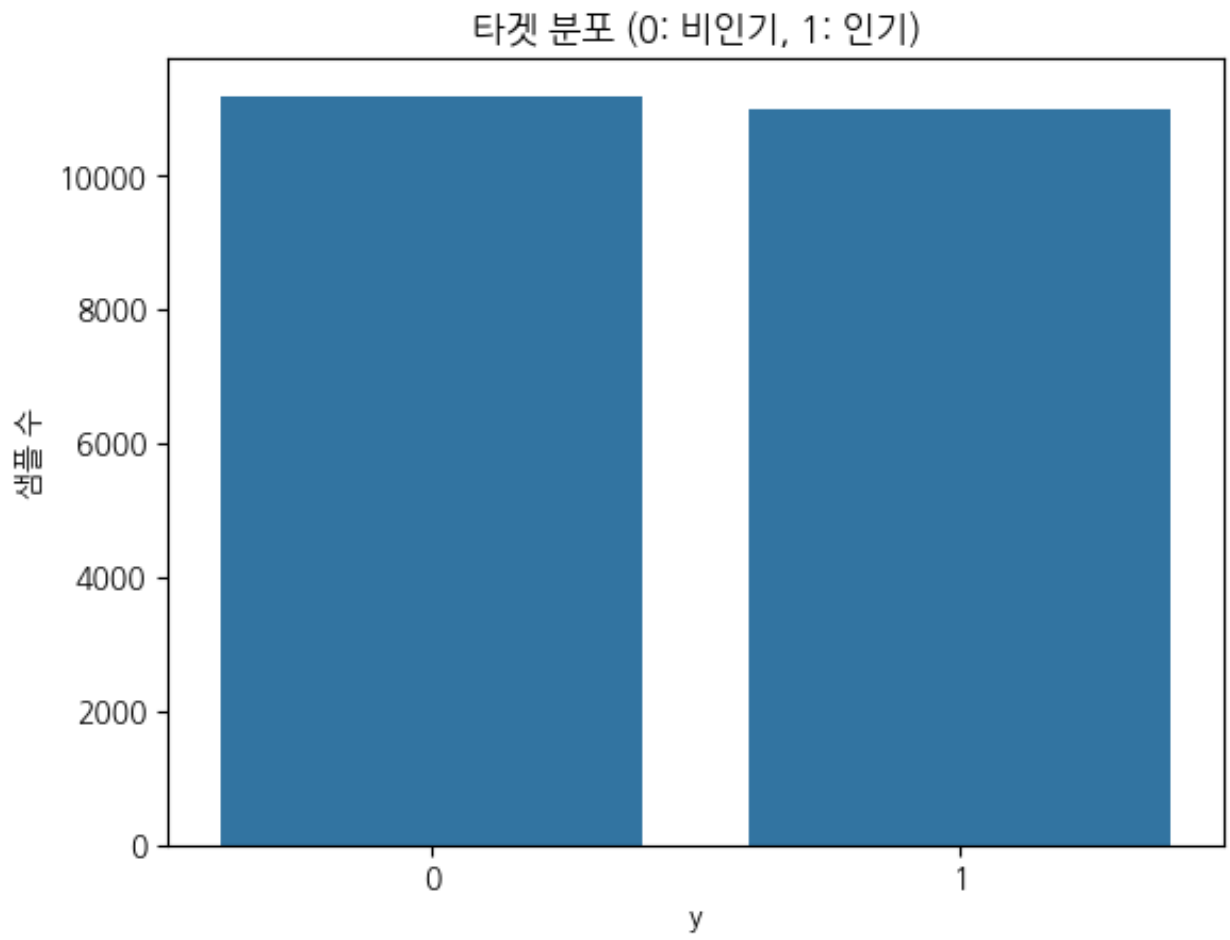
4. 수치형 변수 분포

수치형 변수 대부분이 왜도가 있어서, 즉 비대칭적이라서 정규화가 필요하다. 로그 스케일 또는 RobustScaler 사용을 고려할 수 있다.



5. 타겟(y) 분포

타켓(y)의 분포는 0: 50.4%, 1: 49.6%으로, 거의 균형을 이루고 있다. 따라서 SMOTE 등 샘플링이 필수는 아니다.



---

6. 상관관계 분석

6-1. 타겟과의 상관관계

y 와의 상관관계(피어슨 상관계수)가 높은 변수 (Top 10)는 다음과 같다.

1. kw_avg_avg (0.161)
2. LDA_02 (-0.148)
3. LDA_04 (0.090)
4. kw_min_avg (0.088)
5. num_hrefs (0.084)
6. kw_max_avg (0.079)
7. LDA_01 (-0.076)
8. num_keywords (0.070)
9. LDA_00 (0.068)
10. global_sentiment_polarity (0.064)

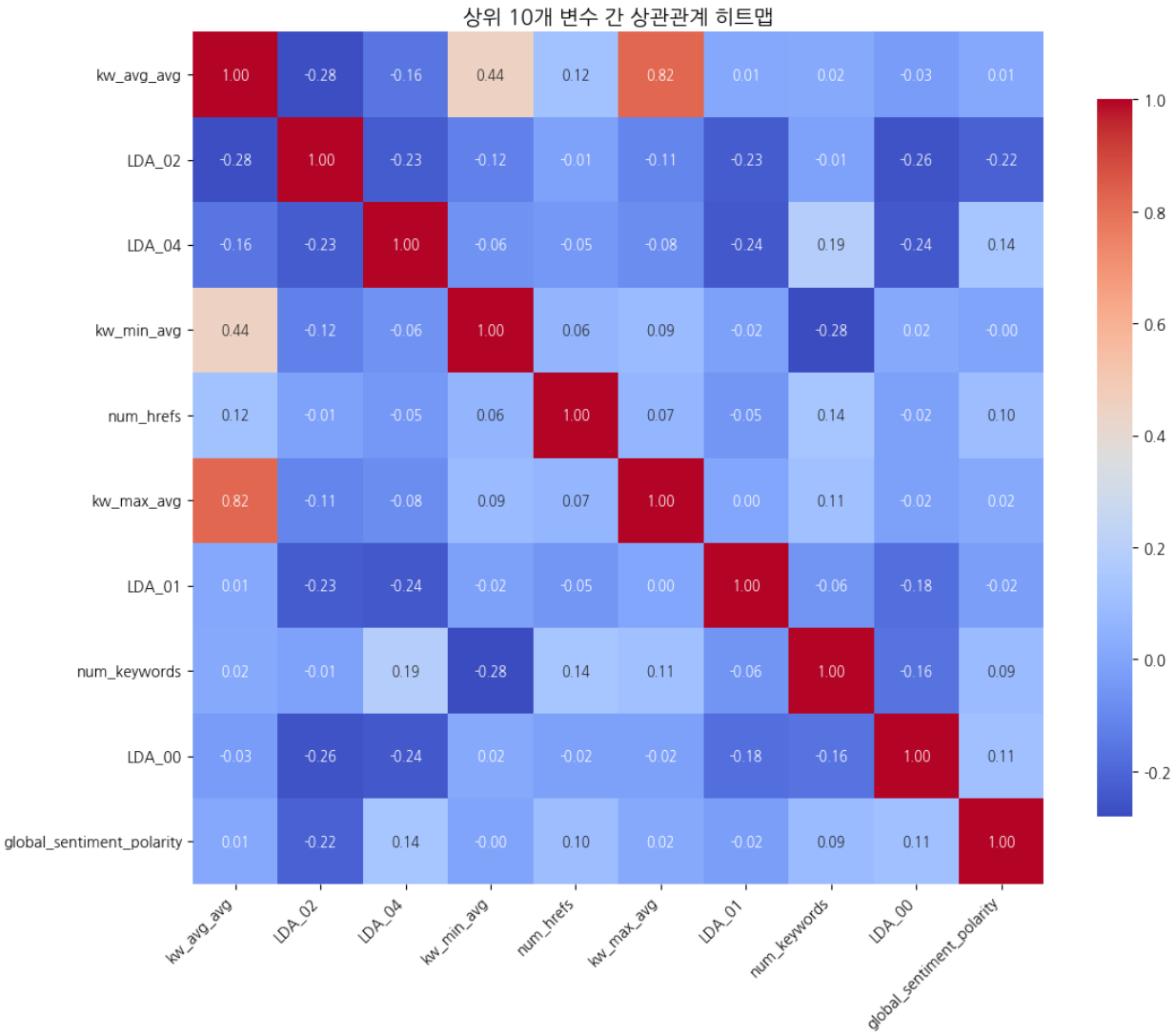
타겟과 상관계수가 높은 수치형 변수를 확인하는 이유는 어떤 변수가 타겟 예측에 유용한 정보를 가지고 있는지 확인하기 위함이다. 상관계수의 절댓값 기준으로 |0.00-0.10|은 거의 무관 또는 무시 가능 수준이고, |0.10-0.30|은 약한 상관관계이다. 타겟과 가장 높은 상관계수를 가지는 kw_avg_avg 변수는 0.161 로 사실상 모든 변수가 타겟과 약한 상관관계 또는 무시 가능한 수준의 상관관계를 가진다. 따라서 단일 변수로는 예측력이 크지 않지만 여러 약한 변수들이 함께 작용하면 좋은 모델이 될 수 있다.

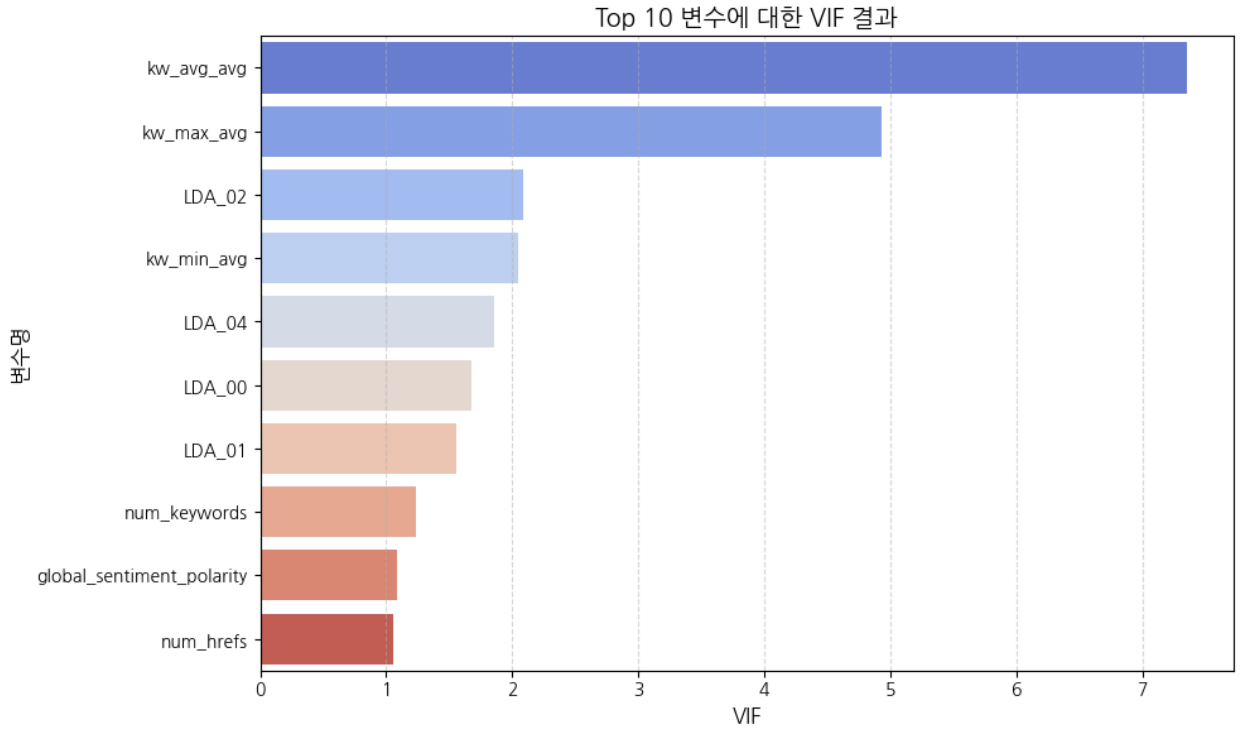
6-2. 변수 간 상관관계

다음으로 입력 변수간 다중공선성 확인을 위해 상관계수 히트맵과 VIF(Variance Inflation Factor, 분산 팽창 계수; 다중공선성을 수치적으로 판단하기 위한 지표)로 타겟과 상관관계가 높은 상위 10 개 변수간 상관관계를 관찰하였다. 다중공선성이 높은 경우 모델이 선형 계열 (로지스틱 회귀, 선형 회귀 등)이면 계수 추청의 불안정성을 유발하여 제거하는 것이 좋다. 그러나 트리 기반의 모델 (랜덤포레스트, XGBoost, LightGBM 등)이면 중복 변수를 포함해도 성능에 큰 영향이 없고, 예측 기여도가 작아 모델이 알아서 무시할 수 있고, 특징 추출 단계에서 상호작용 효과를 위한 보조 변수로 활용이 가능하므로 다중공선성이 높은 변수라도 타겟과의 상관관계가 약하다면 해당 변수를 굳이 제거하지 않아도 된다.

관찰 결과 몇몇 변수들의 다중공선성이 높았다. 예를 들어, kw_avg_avg 와 kw_max_avg 간 상관계수는 0.82, 각각의 VIF 값은 7.35 와 4.9(VIF 값이 5.0-inf 사이이면 다중공선성 의심-심각한

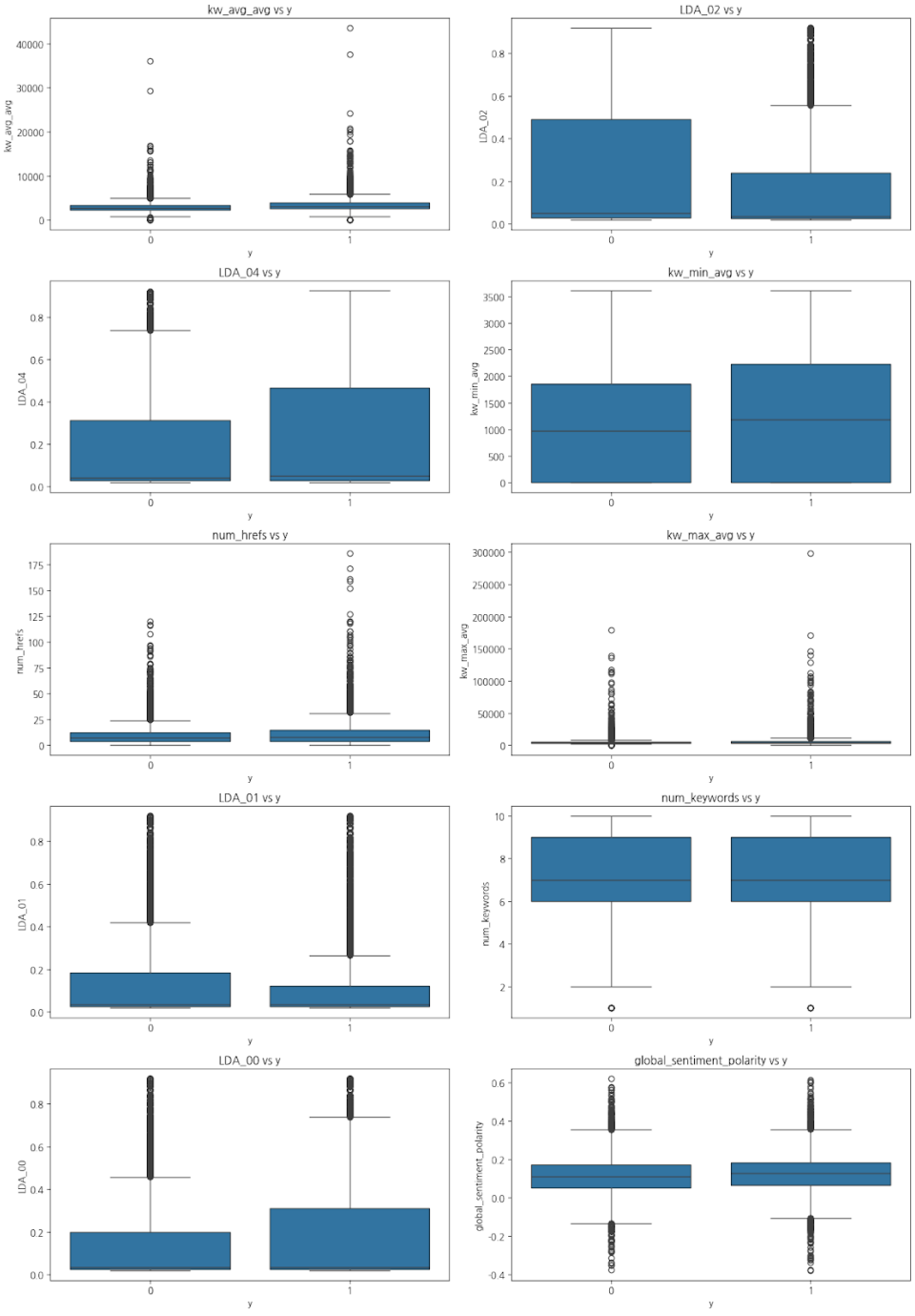
다중공선성이다)이었다. 그러나 결론적으로 모든 변수가 타겟과 무시할 수 있는 수준의 상관관계를 가지므로 트리 기반 모델을 사용한다면 굳이 제거할 필요가 없다고 판단했다.




Top 10 변수에 대한 VIF 결과
kw_avg_avg
kw_max_avg
LDA_02
kw_min_avg
LDA_04
LDA_00
LDA_01
num_keywords
global_sentiment_polarity
num_hrefs
변수명
0
1
2
3
4
5
6
7
VIF

7. 이상치 탐색

이상치 탐색 결과 타겟에 따라 분포 차이 큰 변수 존재했다. 특히 kw_avg_avg, LDA_02, kw_min_avg 등은 outlier 민감한 것으로 확인되었다. 이 경우 수치형 변수 이상치는 IQR 또는 z-score 로 제거/대체를 고려할 수 있다.



3. 데이터 전처리 및 피처 엔지니어링

---

1. 데이터 정리 및 전처리

1-1. 데이터 로드 및 불필요한 컬럼 제거

먼저, 학습 데이터와 테스트 데이터를 각각 불러온 뒤, 분석 및 모델링에 필요하지 않은 컬럼을 제거했다.학습 데이터에서는 'shares'와 'ID' 컬럼을 삭제했고, 테스트 데이터의 경우에는 ID 컬럼을 유지했다. 이렇게 함으로써, 예측 성능에 직접적인 기여를 하지 않는 변수들을 제거하여 모델의 복잡성을 낮출 수 있었다.

1-2. 수치형 변수 이상치 처리 (로그 변환)

데이터셋 내의 일부 수치형 변수들은 이상치가 존재하거나 분포의 왜도가 큰 것을 발견했다. 이를 해결하기 위해 다음 변수들에 대해 로그 변환을 수행했다. np.log1p 를 사용하여 음수값이 없도록 처리하고, 데이터의 스케일을 안정화시켜 모델 학습의 수렴 속도를 높였다.

‘num_hrefs’,‘num_self_hrefs’,‘num_imgs’,‘num_videos’,‘self_reference_avg_sharess’,‘num_hrefs’,’kw_min_min’,’kw_max_min’,’kw_avg_min’,’kw_min_max’,’kw_max_max’,’kw_avg_max’,’kw_min_avg’,’kw_max_avg’,’kw_avg_avg’

1-3. 결측치 처리

다음으로, 수치형 변수와 범주형 변수로 나눈 뒤 각각의 결측치를 처리했다. 수치형 변수는 중앙값(median)으로 결측치를 대체했고, 범주형 변수는 최빈값(mode)으로 결측치를 대체했다. 이렇게 처리된 값은 테스트 데이터셋에도 동일하게 적용하여 학습과 테스트 데이터 간의 일관성을 유지했다.

1-4. 수치형 변수 정규화

수치형 변수는 변수마다 단위나 스케일이 달라서, 경사하강법 기반의 모델에서 학습이 불안정해질 수 있다. 이를 방지하기 위해 각 변수의 평균을 0, 표준편차를 1 로 맞추어 주는 정규화 기법인 StandardScaler 를 사용했다.이를 통해 변수 간의 스케일 차이를 없앴고, 경사하강법의 수렴 속도도 높일 수 있었다. 정규화는 학습 데이터의 통계치(평균, 표준편차)를 기준으로 학습 데이터뿐 아니라 테스트 데이터에도 같은 기준으로 적용하여, 일관성을 유지했다

1-5. 범주형 변수 원핫 인코딩 및 컬럼 정합

범주형 변수는 수치형 모델에서 직접 사용할 수 없기 때문에 OneHotEncoder 를 사용해 원핫 인코딩을 수행했다. 원핫 인코딩은 각 범주를 이진(0/1) 벡터로 변환하는 과정으로, 모델이 각 범주를 독립적인 변수처럼 인식하도록 했다. 하지만 학습 데이터와 테스트 데이터는 서로 범주(컬럼)가 완전히 일치하지 않을 수 있다. 예를 들어 학습 데이터에는 ‘A’, ‘B’, ‘C’가 있었는데, 테스트 데이터에는 ‘A’, ‘C’만 있을 수 있다. 이 경우, 테스트 데이터에 누락된 컬럼을 0 으로 채워 넣어줬다. 이렇게 해서

학습 데이터와 테스트 데이터의 컬럼 순서를 맞춰주고, 모델이 테스트 데이터에서 오류 없이 예측할 수 있도록 했다.

---

2. 파생 변수 생성

2-1. 파생 변수 생성 함수 정의

기존 데이터로부터 추가적인 패턴을 포착할 수 있는 방안을 탐색했다. 탐색 끝에 '파생 변수'를 만드는 방안을 도출했다. 파생 변수는 기존 데이터로부터 새롭게 만들어낸 변수로 기존 변수들로는 표현하지 못하는 숨겨진 패턴을 잡기 위해 사용된다. 파생 변수는 기존 정보의 관계, 비율, 특성을 모델이 더 잘 이해할 수 있도록 돕는다. 반드시 필요한 작업은 아니지만 파생 변수 적용시 더 좋은 성능이 나올 것이라 기대했고, 다음과 같은 파생 변수 생성 함수를 정의했다.

'keyword_ratio': num_keywords 의 개수를 n_tokens_content 로 나눈 비율로, 콘텐츠의 길이에 비해 키워드가 얼마나 많은지를 나타낸다. 키워드가 본문에 자주 나올수록 주제와 관련성이 높고, 기사의 정보 밀도가 높을 가능성이 크다.

'token_density': 콘텐츠 길이(n_tokens_content)를 이미지 개수(num_imgs)로 나눈 값으로, 이미지당 얼마나 많은 콘텐츠 토큰이 있는지를 나타낸다. 콘텐츠가 긴데 이미지가 적으면 정보량이 많고 딱딱한 기사일 수 있고, 반대로 짧은 글에 이미지가 많으면 시각적으로 친절한 기사일 수 있어 이는 기사 유형 구분, 독자 반응과 관련이 있다고 판단했다.

'href_self_ratio': 자기 자신을 가리키는 하이퍼링크 개수(num_self_hrefs)를 전체 하이퍼링크 개수(num_hrefs)로 나눈 비율이다. 자기 기사나 도메인을 많이 인용하는 경우는 자체 콘텐츠 의존성이 높고, 외부 링크가 많은 기사는 출처 다양성이 높은 기사일 수 있다. 이는 독자의 신뢰도와 신뢰도 유형 판단에 힌트가 될 것이라 판단했다.

'stopword_diversity': 불용어 중 고유한 토큰의 개수(n_non_stop_unique_tokens)를 불용어의 전체 개수(n_non_stop_words)로 나눈 비율로, 불용어 토큰의 다양성을 측정한다.고유 단어가 많다는 건 어휘의 다양성과 복잡성이 높다는 의미로 지적이고, 심층적으로 분석된 기사일 확률이 높다는 의미와 같다.

'keyword_score_max': 키워드별 최대 점수(kw_max_max)를 키워드 개수(num_keywords)로 나눈 값으로, 특정 문서에서 가장 중요한 키워드가 얼마나 큰 점수를 받았는지를 나타낸다. 문서의 핵심 키워드가 주제나 내용의 중심을 잘 잡아주는 정도를 반영하며, 이는 문서의 핵심성과 관련성이 높을 가능성이 있음을 시사한다.

'keyword_score_avg': 키워드의 평균 점수(kw_avg_avg)를 키워드 개수(num_keywords)로 나눈 값으로, 문서의 전반적인 키워드 품질을 나타낸다. 키워드들이 고르게 중요도가 높은 문서는

전반적으로 주제가 잘 정리된 문서일 가능성이 있고, 독자가 정보를 쉽게 파악할 수 있을 것으로 판단했다.

'keyword_contrast':키워드의 최대 점수(kw_max_max)와 최소 점수(kw_min_min)의 차이로, 키워드 간의 편차나 대조 정도를 나타낸다. 대조가 큰 문서는 특정 키워드에 집중도가 높아져 주제를 명확히 드러낼 가능성이 있고, 반대로 대조가 작은 문서는 정보가 고르게 분포돼 있다는 특징을 가질 수 있다고 기대했다.

함수는 기존의 데이터프레임을 복사하여 위의 계산을 수행한 후, 7 개의 파생 변수를 새로 추가해 반환하도록 구성했다.

2-2. 파생 변수 적용

정의된 파생 변수 생성 함수를 학습, 검증, 테스트 데이터셋 각각에 적용했다. 이를 통해 모든 데이터셋에 동일한 파생 변수가 추가되어, 이후 모델 학습 및 평가에서 일관성을 유지할 수 있도록 했다.

적용된 데이터셋:

- X_train: 최종 학습용 데이터셋 (훈련 데이터)
- X_val: 모델 성능 검증을 위한 검증용 데이터셋
- X_test: 실제 테스트/예측에 사용되는 데이터셋

---

3. 데이터셋 분할 (훈련/검증)

훈련 데이터를 8:2 비율로 분할하였으며, 클래스 비율을 유지하여 모델이 클래스 불균형의 영향을 덜 받도록 하였다. 이를 통해 실제 배포 환경과 유사한 데이터 분포를 유지하며 검증 데이터셋의 평가 신뢰도를 높였다.

4. 모델링 및 학습

본 프로젝트에서는 Gradient Boosting 계열의 개별 모델(LightGBM, XGBoost, CatBoost)을 결합한 스태킹(Ensemble) 모델을 설계하고 학습하였다.

학습 데이터의 클래스 불균형 문제를 해결하기 위해 SMOTE(Synthetic Minority Over-sampling Technique)를 적용하였으며, F1 Score 최적화 기반의 임계값 탐색을 통해 모델의 성능을 개선하였다.

---

1. 최적 임계값

모델 학습에서 기본 임계값(0.5)은 모든 경우에 최적이라고 보기 어렵다. 클래스 불균형 데이터에서는 F1 Score 가 성능 평가 지표로 적합하므로, 훈련 데이터셋에 다양한 threshold(임계값)를 적용하고 F1 Score 를 계산하여 최적 임계값을 탐색하였다. threshold 는 0.1~0.9 까지 0.01 간격으로 설정하였으며, 각 임계값별 F1 Score 를 계산하여 성능이 가장 높은 최적 임계값(best_threshold)을 도출하였다.

---

2. 사용한 모델 개요

2-1. Base learners (LightGBM, XGBoost, CatBoost)

본 프로젝트에서는 Gradient Boosting 계열의 세 가지 모델을 Base Learner 로 사용하였다.

- **LightGBM(LGBMClassifier)**: 빠른 학습 속도와 효율적인 메모리 사용을 자랑하며, 대규모 데이터 처리에 적합한 Gradient Boosting 모델
- **XGBoost(XGBClassifier)**: Regularization(정규화) 및 병렬 학습 기능을 통해 과적합 방지와 안정적인 성능을 제공하는 모델
- **CatBoost(CatBoostClassifier)**: 범주형 변수 자동 처리 기능과 높은 예측 성능을 제공하는 Gradient Boosting 모델

2-2. Stacking 모델 구조

- Base Learner 들의 예측 결과와 원본 피처를 함께 메타모델에 전달하는 StackingClassifier 를 구현하였다.
- passthrough=True 옵션을 적용하여 Base Learner 의 출력값과 원본 데이터를 모두 활용함으로써 메타모델의 학습력을 강화하고, 다양한 피처 기반 예측을 가능하게 하였다.

2-3. 메타 모델(XGBoost)

최종적으로 스태킹 모델의 메타모델로는 XGBoost 를 사용하였다. XGBoost 는 안정적인 성능과 과적합 방지 효과를 제공하며, 최적의 하이퍼파라미터(n_estimators=100, max_depth=3, learning_rate=0.1 등)를 적용하여 메타 단계에서 최종 예측을 수행하였다.

---

3. 클래스 불균형 처리 방법 - SMOTE 오버샘플링

본 프로젝트에서는 SMOTE(Synthetic Minority Over-sampling Technique)를 적용했다.

- SMOTE 는 소수 클래스 샘플 간의 거리를 기반으로 새로운 데이터를 합성하여 소수 클래스의 비율을 증가시킨다.
- 이로 인해 학습 데이터의 클래스 비율을 균형 있게 맞추고, 모델의 학습 편향을 줄였다.
- 결과적으로 Recall 과 Precision 의 균형을 맞추고 F1 Score 를 향상시킬 수 있었다.

---

4. 모델 학습 과정

학습 데이터셋은 전처리(불필요 컬럼 제거, 로그 변환, 결측치 대체, 정규화, 원-핫 인코딩)를 마친 후 파생 피처를 추가해 학습의 정보량을 늘렸다.

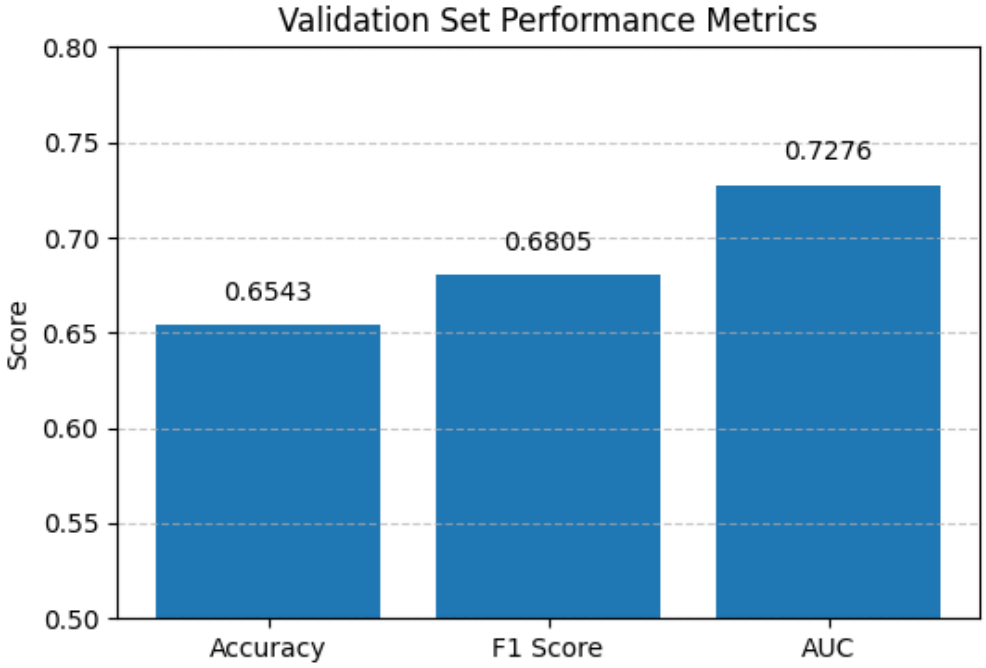
- SMOTE 로 오버샘플링된 데이터셋을 기반으로 Base Learner(LGBM, XGBoost, CatBoost)와 Meta Model(XGBoost)로 구성된 StackingClassifier 를 학습했다.
- 학습은 fit 메서드를 사용하여 오버샘플링된 데이터를 학습에 투입했다.
- Validation 데이터셋에 대해서는 predict_proba 메서드로 확률 예측을 수행하고, 임계값(best_threshold)을 적용하여 최종 이진 분류(0,1)를 수행했다.
- 테스트 데이터셋에도 동일한 방식으로 예측을 수행하고, 예측값(y), 예측확률(y_prob), ID 를 포함한 prediction.csv 파일을 생성했다.

5. 결과 및 성능 평가

---

1. 예측 결과

본 프로젝트에서 구축한 모델의 성능은 내부 검증 데이터셋을 기준으로 수행되었으며, 다음의 세 가지 대표적인 분류 지표를 사용하였다. 정확도(Accuracy), F1 Score, AUC (Area Under Curve)가 사용되었으며, 각 각 다음과 같은 값을 기록하였다.



- 정확도 (Accuracy) : 0.6543
- F1 Score : 0.6805
- AUC (Area Under Curve) : 0.7276

본 프로젝트에서는 모델 성능의 전반적인 균형을 평가하기 위해, Accuracy, F1 Score, AUC 의 평균값을 종합 평가 지표로 정의하였다. 해당 계산식은 다음과 같다.

$$\text{Main Evaluation Metric} = \frac{0.6543 + 0.6805 + 0.7276}{3} = 0.6875$$

이러한 평균 점수는 모델이 특정 지표에 과도하게 편중되지 않고, 균형 잡힌 예측 성능을 보였음을 나타낸다.

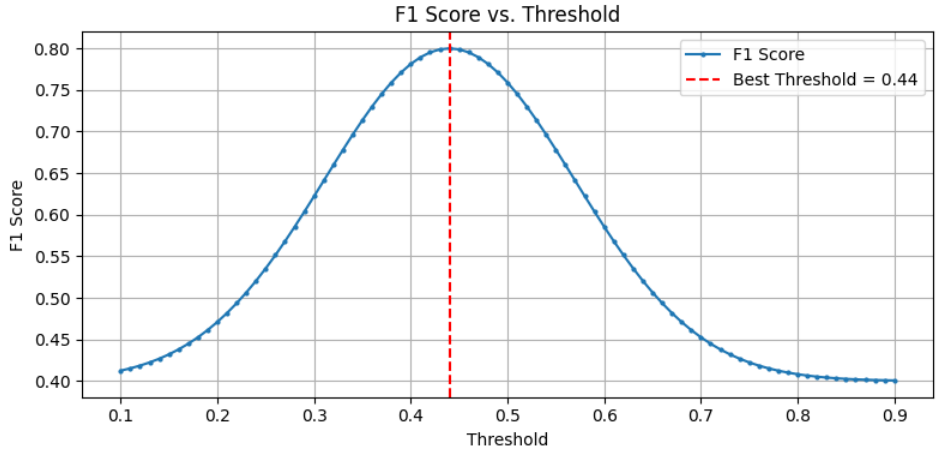
---

2. 최적 임계값 설정 (Threshold Tuning)

모델 학습 이후, 훈련 데이터에 대해 다양한 임계값(threshold)을 적용하고, 각 threshold 별로 F1 Score 를 계산하여 성능이 가장 높은 임계값을 탐색하였다. 그 결과, F1 Score 가 최대가 되는 최적 임계값은 0.44 로 도출되었으며, 이 값을 기준으로 검증 및 테스트 데이터셋에 대한 최종 예측이 수행되었다.

- 최적 임계값 (threshold) : 0.44
- 해당 임계값에서의 훈련 F1 Score : 0.7998

이는 모델이 클래스 불균형 상황에서도 민감도(recall)와 정밀도(precision) 간의 균형을 효과적으로 조정할 수 있었음을 보여준다.



---

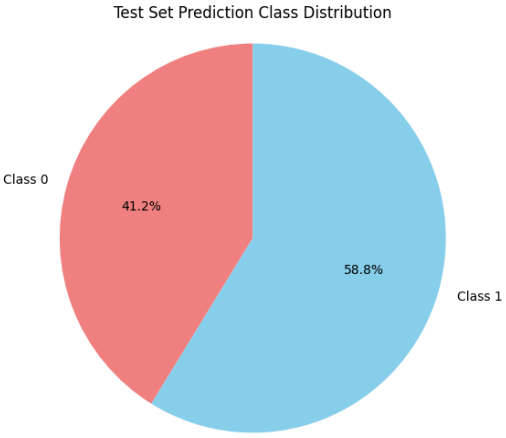
3. 결과 해석

Stacking 기반 앙상블 모델은 단일 모델에 비해 전반적으로 안정적이고 강건한 성능을 보였다. 특히 F1 Score 과 AUC 에서 우수한 결과를 기록하며, 불균형 데이터에서도 높은 판별력을 유지하였다. 또한, threshold 최적화를 통해 정밀도와 재현율 사이의 균형을 실질적으로 조절함으로써, 현실적인 이진 분류 문제에 적용 가능한 수준의 예측 성능을 확보하였다.

---

4. 테스트 세트 예측 클래스 분포

테스트 데이터셋에 대한 최종 예측 결과를 분석한 결과, 전체 9,515 개의 샘플 중 5,595 개(58.8%)가 클래스 1, 3,920 개(41.2%)가 클래스 0 로 분류되었다. 이는 원본 데이터에 존재하는 클래스 불균형을 고려했을 때, 모델이 단순히 다수 클래스에만 예측을 집중하지 않고 최적화된 임계값(0.44)을 기반으로 적절한 판별 기준을 설정하여, 양 클래스 모두를 안정적으로 예측했음을 의미한다. 또한 이러한 분포는 threshold 조정의 실질적인 효과를 보여주는 결과로, 모델이 실제 운영 환경에서도 신뢰도 있는 이진 분류 결과를 생성할 수 있는 구조임을 시사한다.


Test Set Prediction Class Distribution
Class 0
41.2%
58.8%
Class 1

6. 결론 및 시사점

본 프로젝트에서는 LightGBM, XGBoost, CatBoost 를 기반으로 한 Stacking 앙상블 모델을 설계하고, 성능 개선을 위해 다양한 전처리 기법, 특성 엔지니어링, SMOTE 기반 오버샘플링 및 임계값 최적화(Threshold Tuning)을 적용하였다. 그 결과, 내부 검증 데이터 기준 평균 0.6875 의 성능 지표를 달성하였으며, 특히 불균형 데이터 상황에서도 F1 Score 과 AUC 측면에서 우수한 성능을 나타냈다. 또한, Threshold 최적화를 통해 현실적인 분류 문제에서의 정밀도와 재현율 간 균형을 성공적으로 조율하였고, 테스트 세트에서도 적절한 클래스 분포를 유지함으로써 실제 응용 가능성 높은 모델 구조임을 입증하였다.

이번 실험을 통해, 불균형한 데이터 상황에서는 임계값 조정과 같은 전략이 예측 성능에 중요한 영향을 줄 수 있다는 점을 확인할 수 있었다. 특히 F1 Score 와 AUC 같은 지표를 바탕으로 모델을 평가하고 개선하는 과정이 실제 성능 향상에 도움이 되었다. 향후에는 외부 데이터셋을 활용한 일반화 성능 평가, 다양한 앙상블 구조 실험, 또는 예측 결과를 더 잘 이해할 수 있는 해석 가능한 모델 적용 등으로 실험을 확장해볼 수 있을 것으로 기대된다.