YouTube 인기영상 예측

CodeStates AI 18 류예린

목 차

문제 설정

우리는 유튜브 내 인기 동영상에 광고를 송출하고자 합니다.

그러나 이미 인기가 있는 광고에 송출하기보다,

인기 동영상을 조기에 예측하여 타겟을 좁히고 싶습니다.

따라서 이에 따른 인기 동영상 분류 예측 모델을 제작하고자 합니다.

이는 사람들이 좋아하는 인기 동영상에 광고를 송출하여,

광고 상품의 호감도를 높이고 비용을 절감할 수 있을 것입니다.

인기동영상 분류 예측 모델

가설 설정

가설 1

유튜브 업로드 초반 영상의 정보는
추후 동영상의 인기 여부에 영향을 미치는 특성일 것이다.

가설 1에 대한 세부 내용

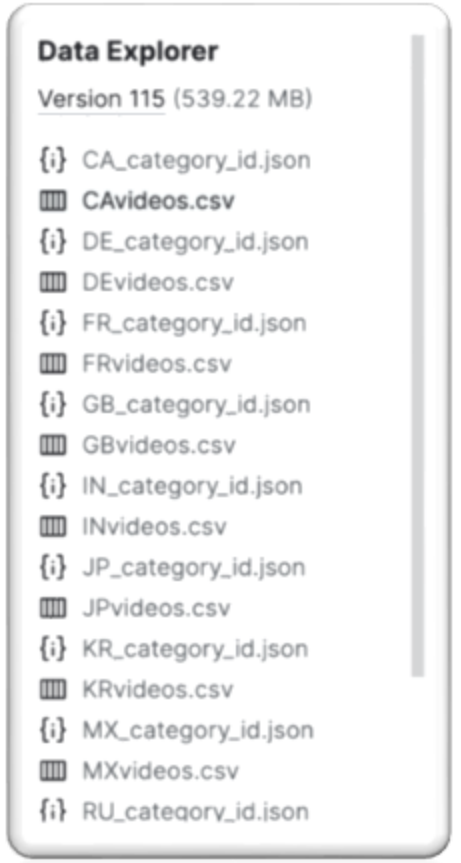
제목의 길이가 짧을 수록 인기 상승
싫어요 비율이 높을 수록 인기 하락
좋아요 비율이 높을 수록 인기 상승
Description이 있으면 인기 상승
태그의 개수가 많을수록 인기 상승

가설 2

분류 예측 모델을 활용하여 category 별 광고 적합성을 파악하고,
광고 송출의 target을 좁혀 보다 효과적인 광고를 기대할 수 있을 것이다.

데이터 셋 설명

https://www.kaggle.com/datasets/datasnaek/youtube-new?select=CAvideos.csv



2006.07 ~ 2018.06 총 10개 국가의
YouTube 인기 동영상(Trending Video)의 정보를 담고 있는 데이터 셋

인기 동영상의 특성을 수집하기에 적합하다 판단

데이터 셋 설명

- 총 16개의 column
- 375942개의 영상 정보

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 375942 entries, 0 to 375941
Data columns (total 17 columns):
 #   Column                  Non-Null Count   Dtype
---  ------                  --------------   -----
 0   Nation                  375942 non-null  object
 1   video_id                375942 non-null  object
 2   trending_date           375942 non-null  object
 3   title                   375942 non-null  object
 4   channel_title           375942 non-null  object
 5   category_id             375942 non-null  int64
 6   publish_time            375942 non-null  object
 7   tags                    375942 non-null  object
 8   views                   375942 non-null  int64
 9   likes                   375942 non-null  int64
 10  dislikes                375942 non-null  int64
 11  comment_count           375942 non-null  int64
 12  thumbnail_link          375942 non-null  object
 13  comments_disabled       375942 non-null  bool
 14  ratings_disabled        375942 non-null  bool
 15  video_error_or_removed  375942 non-null  bool
 16  description             356464 non-null  object
dtypes: bool(3), int64(5), object(9)
memory usage: 41.2+ MB
```

Nation: 국가 코드
title: 영상 제목
category_id: 영상 카테고리 아이디
publish_time: 영상 업로드 날짜, 시간
tags: 영상에 달린 태그
veiws: 조회수
likes: 좋아요 수
dislikes: 싫어요 수
comment_count: 댓글 수
comment_disabled: 댓글창을 닫은 여부
ratings_disabled: 좋아요, 싫어요 공개 여부
video_error_or_removed: 영상 오류, 삭제 여부
description: 영상 설명란

국가 코드는
데이터 셋을 합친 뒤
추가하였음

카테고리 아이디는
국가 별로 다른 ID가
부여되어 있었으나
국가별 .json file에서 불러와
각각 새로 기입하였음

데이터 셋 설명

- **Add features** 영상이 업로드 되고 1시간 이내의 초반 정보로 알 수 있는 특성들을 기반으로 가공

1. publish_time
   - date와 time으로 분리
     - date는 요일로 변환(publish_day)
   - Time을 다섯 column으로 분리
     - 총 합(초 단위), hour, minute, second
2. likes 대비 dislike의 비율
3. views 대비 likes의 비율
4. title의 길이
5. tag의 개수
6. description 여부(1, 0)
7. views대비 comment_count의 비율
8. comments_disabled, ratings_disabled 여부: (1, 0)로 변환

publish_time: 영상 업로드 날짜, 시간
likes: 좋아요 수
dislikes: 싫어요 수
veiws: 조회수
title: 영상 제목
tags: 영상에 달린 태그
description: 영상 설명란
comment_count: 댓글 수
comment_disabled: 댓글창을 닫은 여부
ratings_disabled: 좋아요, 싫어요 공개 여부

데이터 셋 가공

- **Delete features**

1. 정보 누수 feature
   - comment_count
   - likes
   - dislikes
   - views
2. 분석에 필요 없는 feature
   - vedio_id
   - trending_date
   - title
   - channel_title
   - publish_date
   - tags
   - thumbnail_link
   - video_error_or_removed

정보 누수란?
타겟과 인과 관계가 있는 feature을 포함하는 것

이는 지나치게 낙관적인 예측 오류를 범한다.

comment_count: 댓글 수
likes: 좋아요 수
dislikes: 싫어요 수
views: 조회수
vedio_id: vedio의 고유 아이디
trending_date: 인기 동영상에 오른 날짜
title: 영상 제목
channel_title: 채널 이름
publish_date: 영상 업로드 날짜
tags: 영상에 달린 태그
thumbnail_link: 썸네일 링크
video_error_or_removed: 영상 오류, 삭제 여부

타겟 설정

타겟 분류 기준

좋아요 수
상위 30%

조회수
상위 30%

1

likes >= 10000
And
Veiws >= 500000

약 20% 데이터

위에 해당되지
않은 데이터

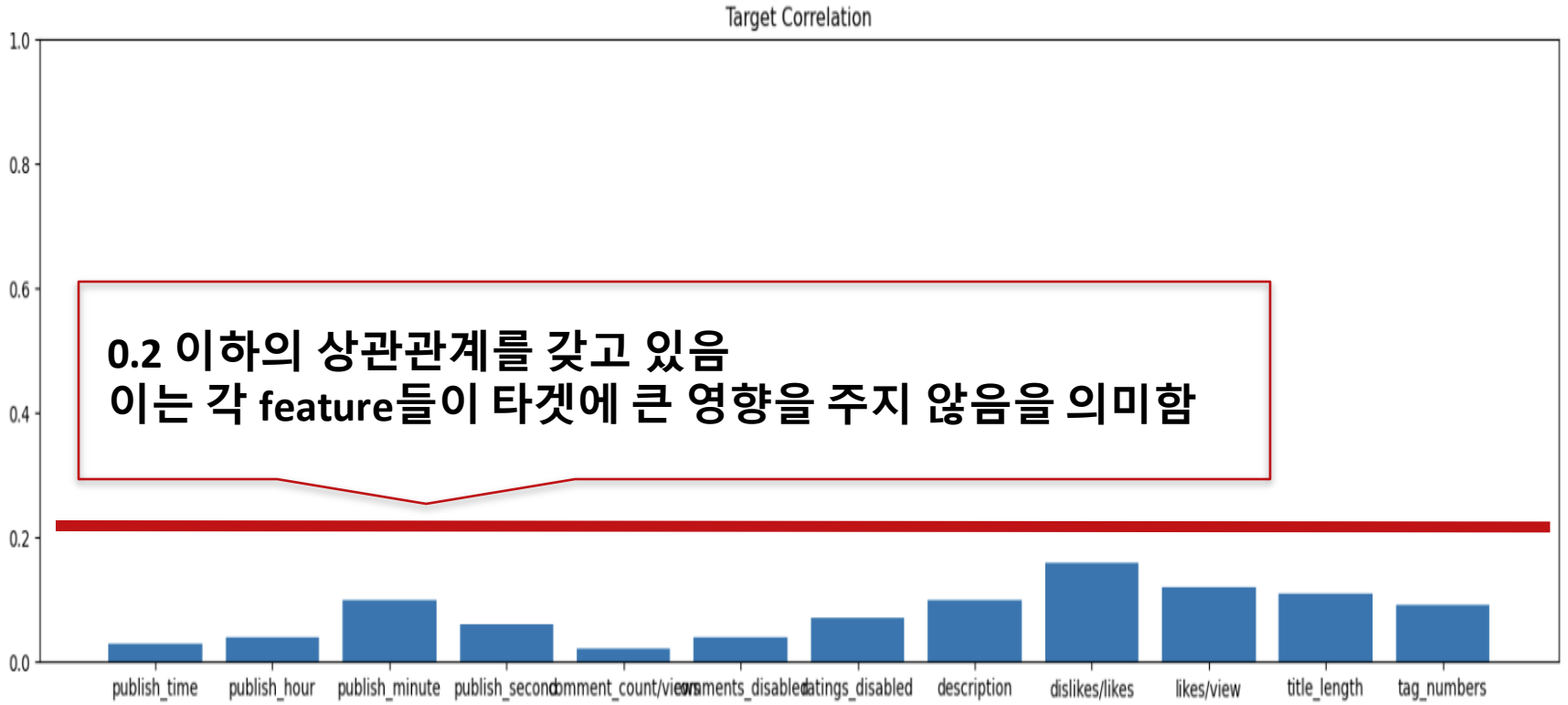
약 80% 데이터

0

타겟 설정

데이터 누수 문제: 타겟 간의 상관 계수가 지나치게 높은 것이 없음을 확인

```
publish_time            0.03
publish_hour            0.04
publish_minute          0.10
publish_second          0.06
comment_count/views     0.02
comments_disabled       0.04
ratings_disabled        0.07
description             0.10
dislikes/likes          0.16
likes/view              0.12
title_length            0.11
tag_numbers             0.09
Name: target, dtype: float64
```



모델링 준비

Train (178,052)

모델을 훈련시키는 데이터 셋

Validation (76,308)

훈련된 모델의 성능을 확인하고 발전시키기 위한 검증 데이터 셋

Test (109,012)

모델의 성능을 최종으로 테스트하기 위한 데이터 셋

```
X_train.shape, y_train.shape
```

```
((178052, 15), (178052,))
```

```
X_val.shape, y_val.shape
```

```
((76308, 15), (76308,))
```

```
X_test.shape, y_test.shape
```

```
((109012, 15), (109012,))
```

모델링 준비

기준 모델

예측을 위해 만든 모델의 기준이 되는 지표

기준 모델의 정확도보다
높은 정확도를 목표로 하기 위함

```python
# base model

from sklearn.metrics import accuracy_score

base = y_train.mode()[0]
baseline = len(y_train) * [base]
baseline_acc = accuracy_score(y_train, baseline)
print(f'기준모델의 정확도: {baseline_acc.round(3)}')
```

```
기준모델의 정확도: 0.799
```

분류 예측 모델은
정확도 80% 이상을 목표로 함

모델링

XGBoost

Tree-based 모델로 복원 추출로 여러 모델을 학습한 후,
모델들의 예측을 합하여 최종 예측을 내는 모델

```
params = {"xgbclassifier__max_depth": [17, 18, 19, 20],
          "xgbclassifier__colsample_bytree": uniform(loc=0.4, scale=0.
         "xgbclassifier__min_child_weight": [6, 7, 8, 9, 10]
         }

f = RandomizedSearchCV(pipe_XGB,
                       param_distributions=params,
                       scoring="f1",
                       n_iter=5,
                       cv=3,
                       verbose=3,
                       random_state=42)
```

```
pipe_XGB = make_pipeline(OrdinalEncoder(),
                         XGBClassifier( objective = "binary:logistic",
                                        eval_metric = "error",
                                        n_estimators = 100,
                                        random_state = 2,
                                        n_jobs = -1,
                                        learning_rate = 0.2,
                                        min_child_weight = 8,
                                        max_depth = 20,
                                        colsample_bytree = 0.55,
                                        gamma = 1,
                                        subsample = 0.5))
```

하이퍼파라미터

모델의 성능을
조절하기 위한 옵션

RandomizedSearch로 하이퍼파라미터 탐색 후 적절히 지정하였음

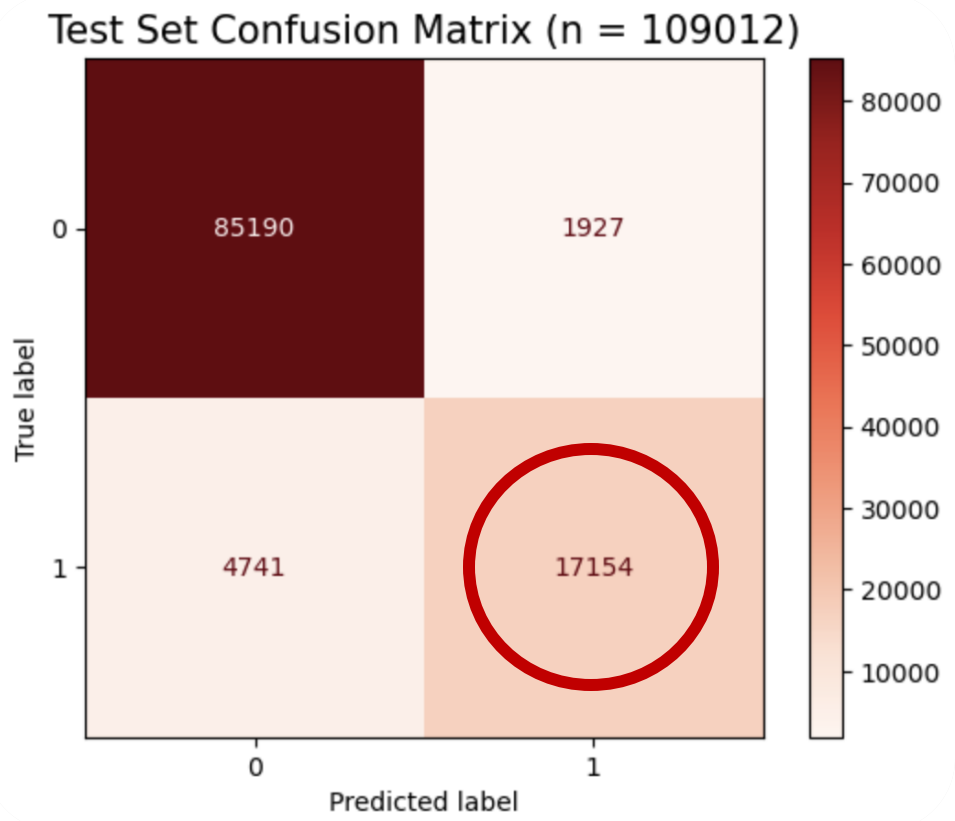
Max_depth: 과적합 조절
Colsample_bytree: 과적합 조절
Min_child_weight: 과적합 조절
Gamma: 과적합 감소 효과
Subsample: 과적합 조절
Learning_rate = 학습률(과적합, 과소적합 조절)

모델링

test report

	precision	recall	f1-score	support
0	0.95	0.98	0.96	87117
1	0.90	0.78	0.84	21895
accuracy			0.94	109012
macro avg	0.92	0.88	0.90	109012
weighted avg	0.94	0.94	0.94	109012

타겟=1



precision

190801개의 1을 예측,
이 중 17154개가 정답

정밀도 90%

recall

21895개의 1 중
17154개를 예측

재현율 78%

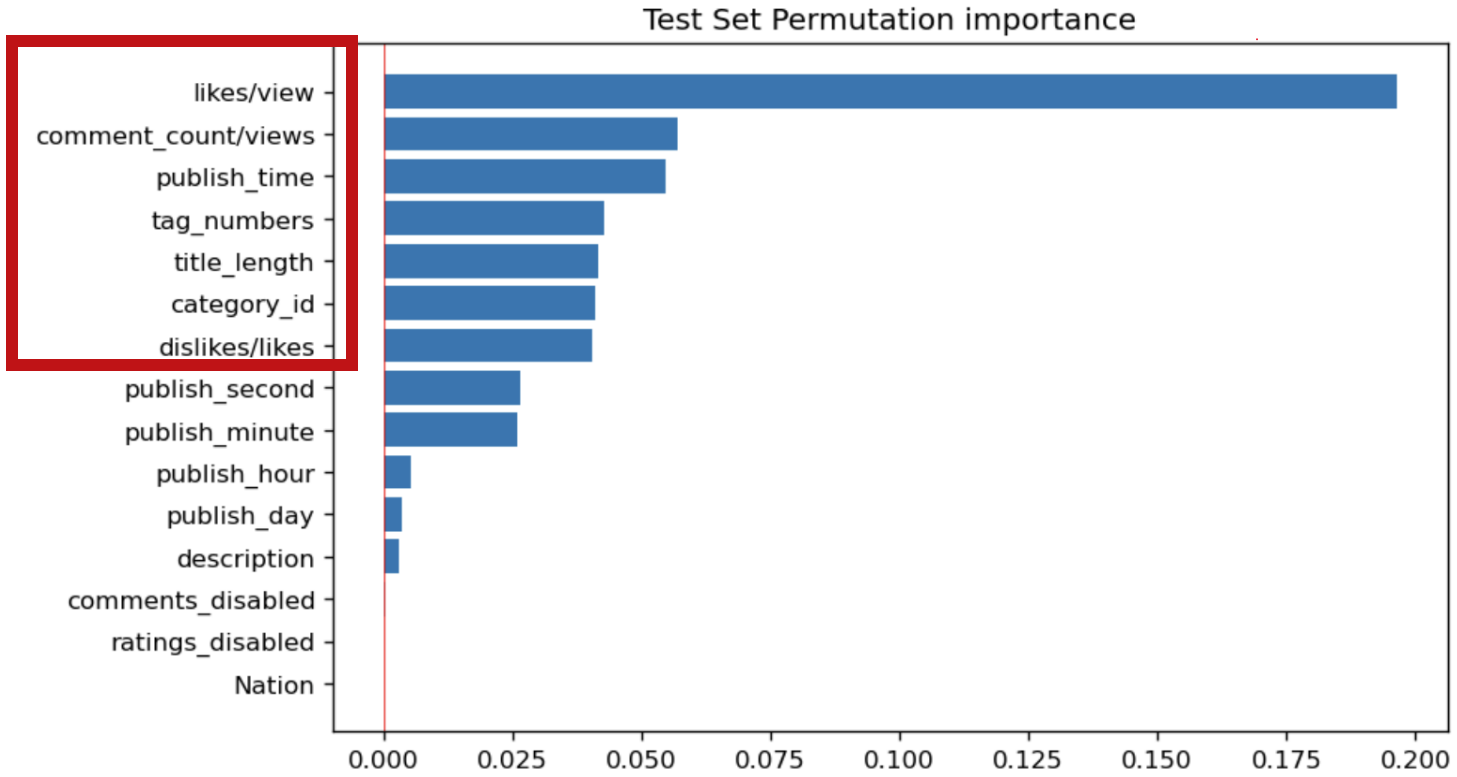
accuracy

108012개의 데이터 중
102344개의 예측이 정답

정확도 94%

모델 해석: Permutation Importance

순열 중요도란?
해당 특성을 무작위로 섞어 제기능을 못하게 만든 후 모델을 만들어 성능을 비교한 후 중요도를 계산한 것



순열 중요도 순위

1. 좋아요 비율
2. 댓글 비율
3. 업로드 시간
4. 태그 개수
5. 제목의 길이
6. 싫어요의 비율

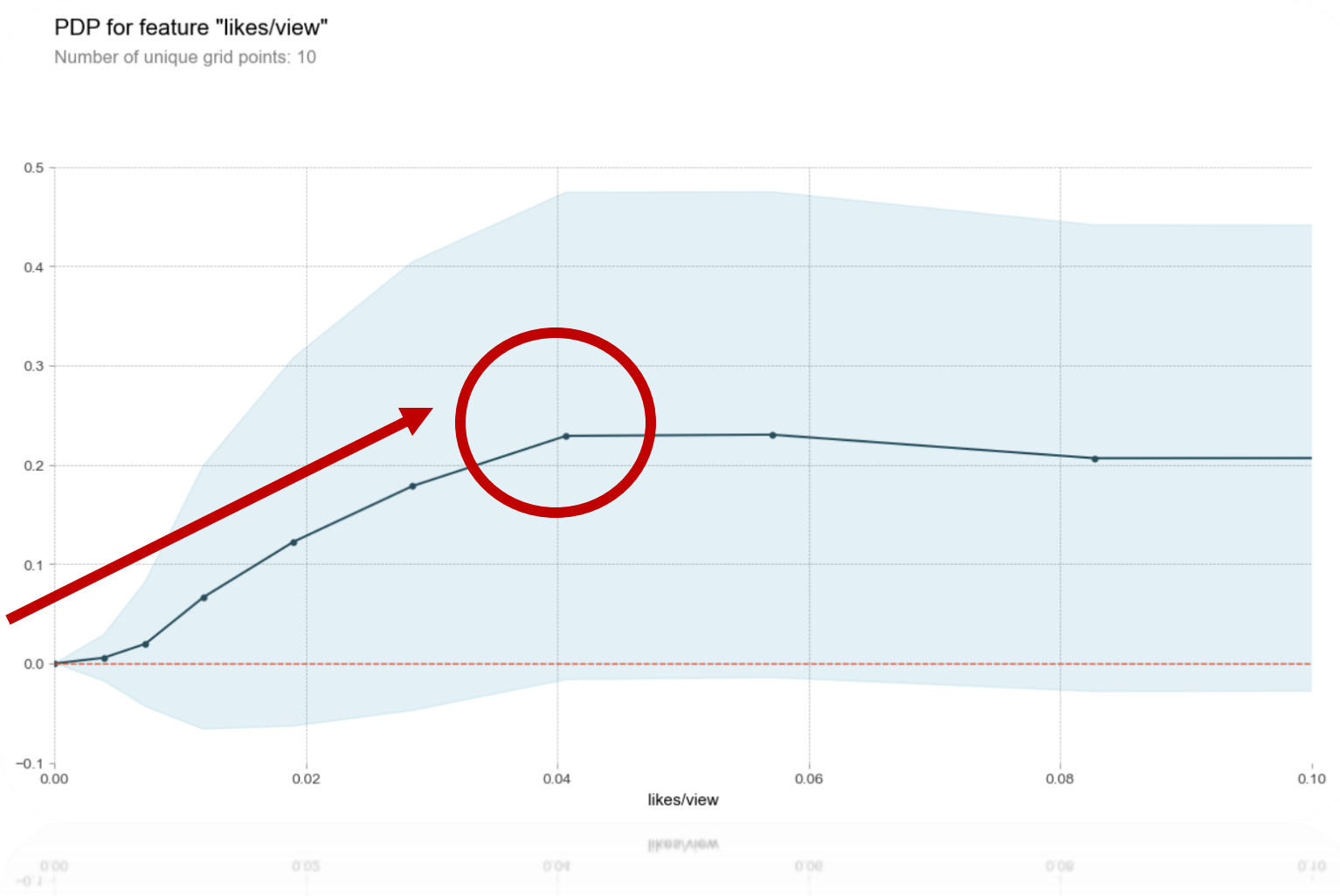
모델 해석: PDP PLOT

Likes/views

조회수 대비 좋아요의 수가
0.04일 때 모델은
20% 이상 확률로
Target을 1로 예측

PDP(Partial Dependence Plots)

개별 특성이 타겟에 어떻게 작용하는지
알아볼 수 있는 도구
model이 x축 지점에서 target이
1일 확률을 y축으로 표시합니다.



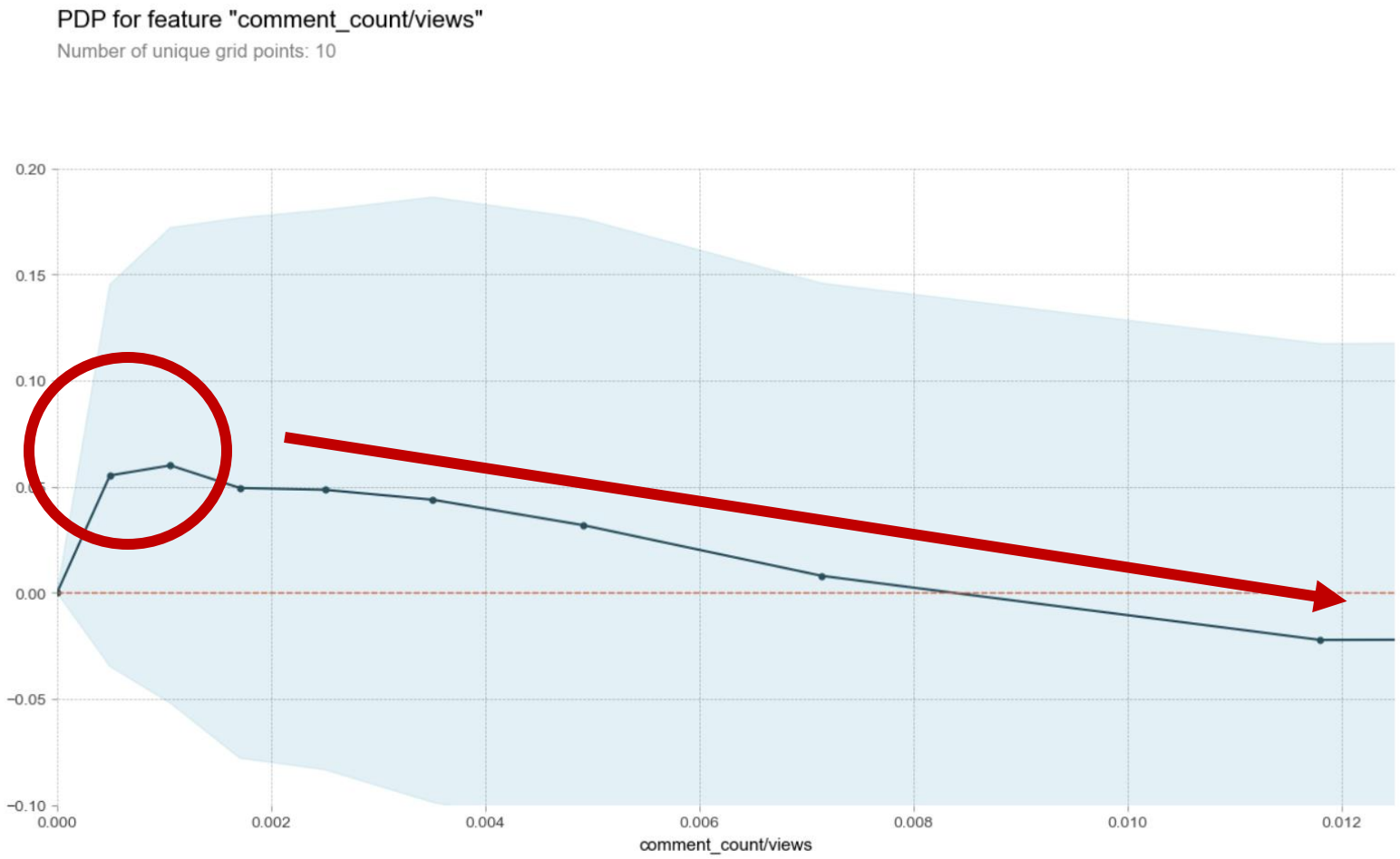
모델 해석: PDP PLOT

comment_count/views

댓글 수가 조회수에 비해 너무 많아도 모델은 Target을 0으로 예측함에 더 가까워짐

PDP(Partial Dependence Plots)

개별 특성이 타겟에 어떻게 작용하는지 알아볼 수 있는 도구
model이 x축 지점에서 target이 1일 확률을 y축으로 표시합니다.



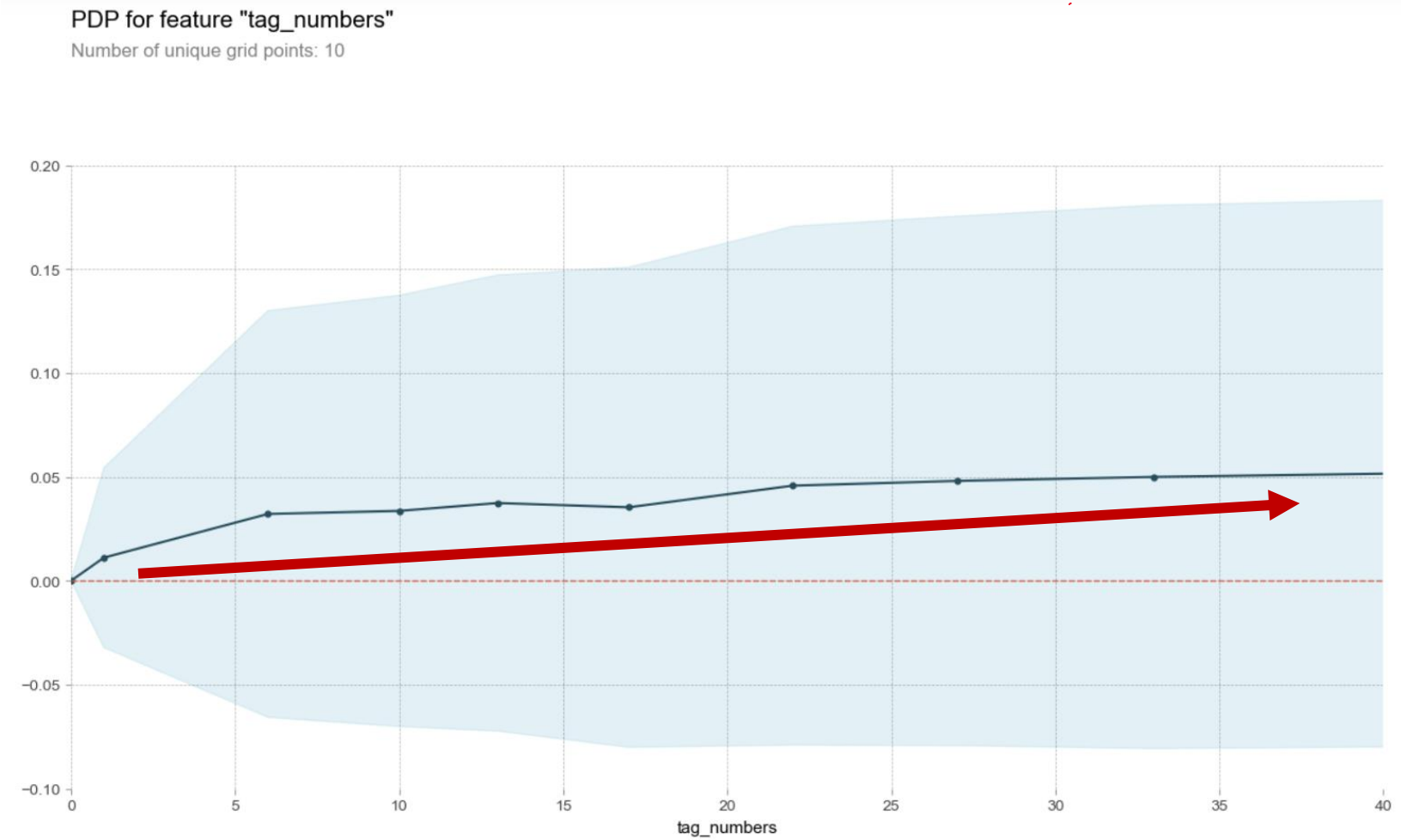
모델 해석: PDP PLOT

tag numbers

Tag 개수가 많아질수록 모델은 미세하게 Target을 1로 예측할 확률이 높아짐

PDP(Partial Dependence Plots)

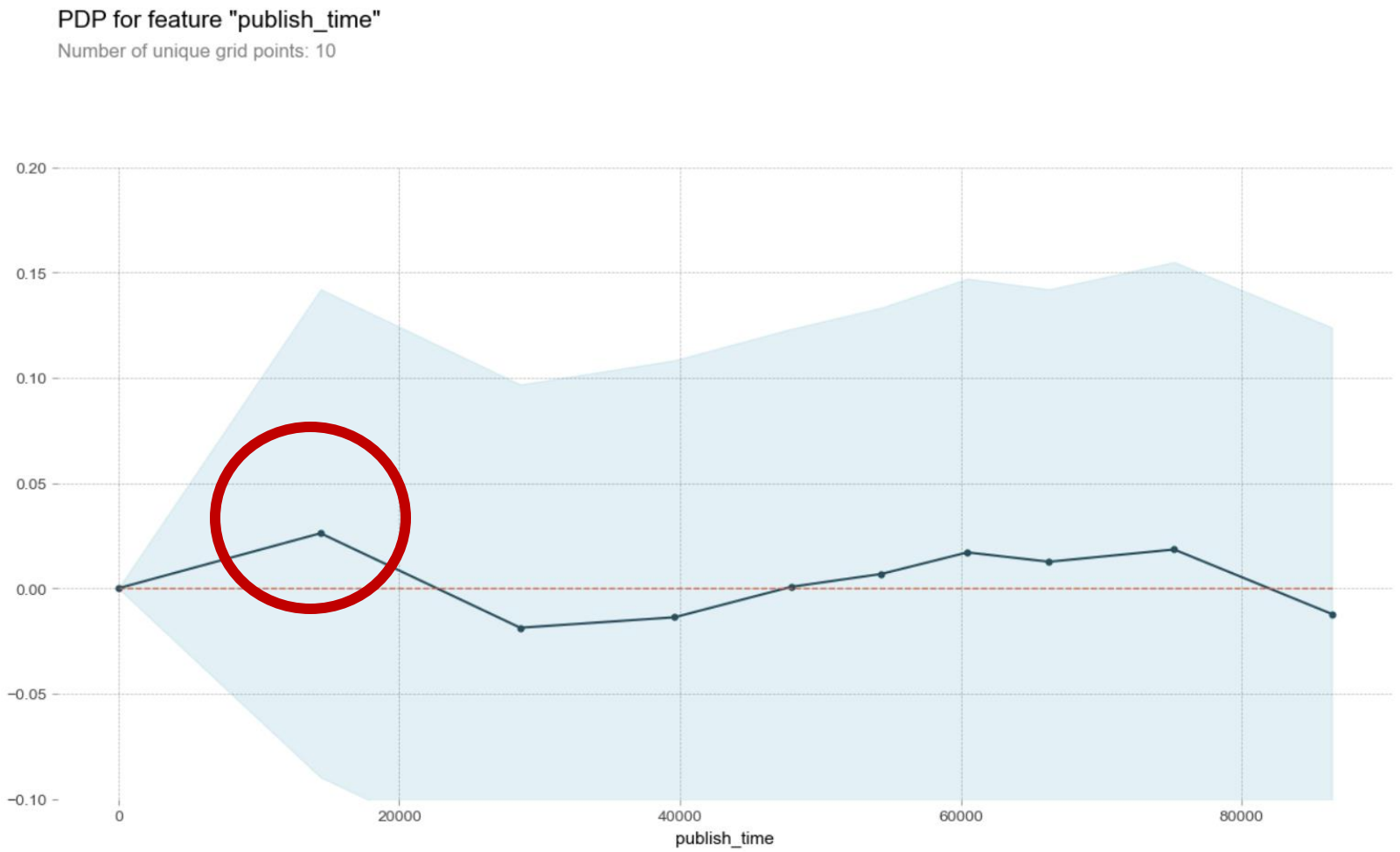
개별 특성이 타겟에 어떻게 작용하는지 알아볼 수 있는 도구
model이 x축 지점에서 target이 1일 확률을 y축으로 표시합니다.



모델 해석: PDP PLOT

publish_time

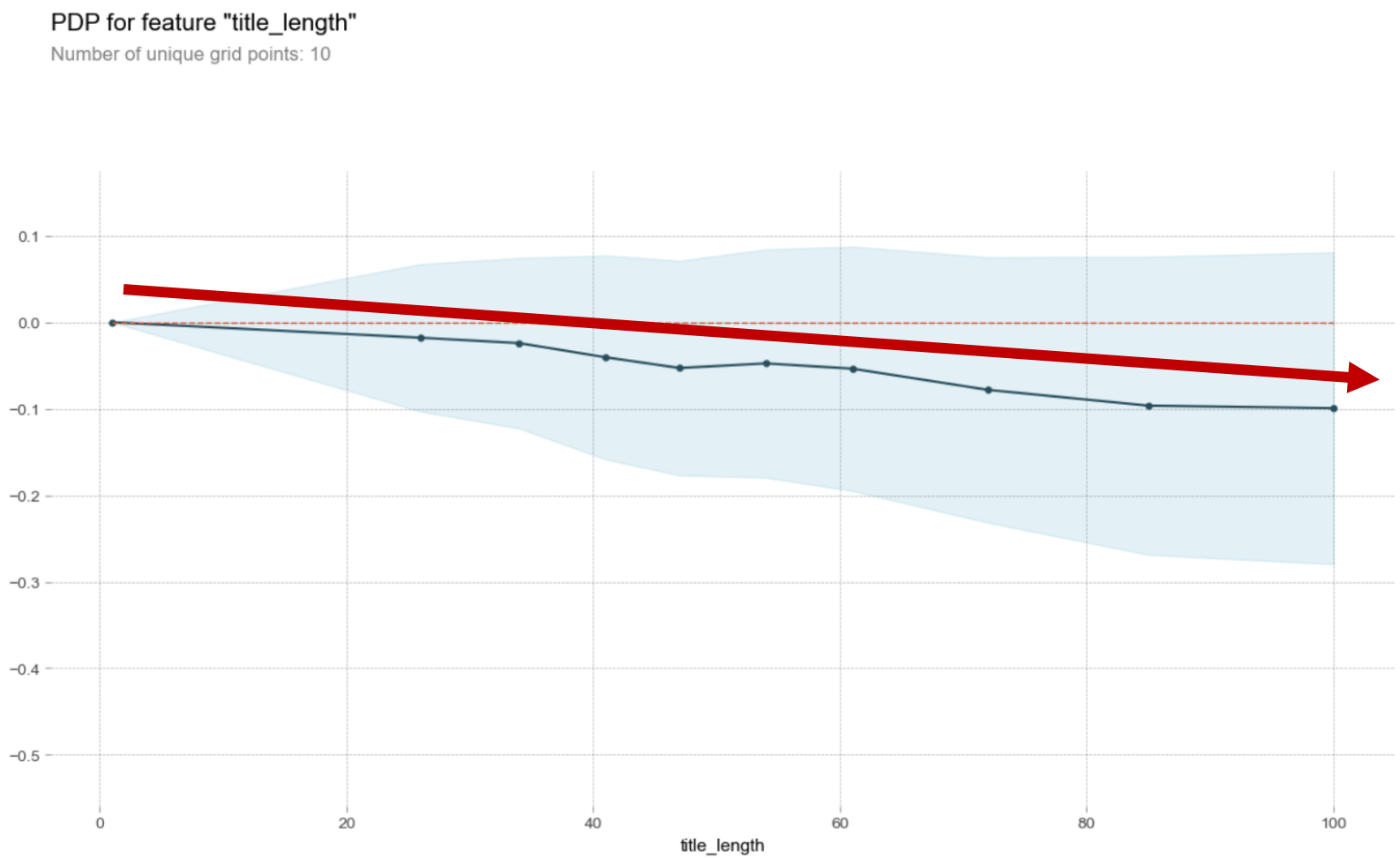
약 새벽 4-5시 사이에 올라온 영상의 target이 1일 확률이 가장 높음



모델 해석: PDP PLOT

title_length

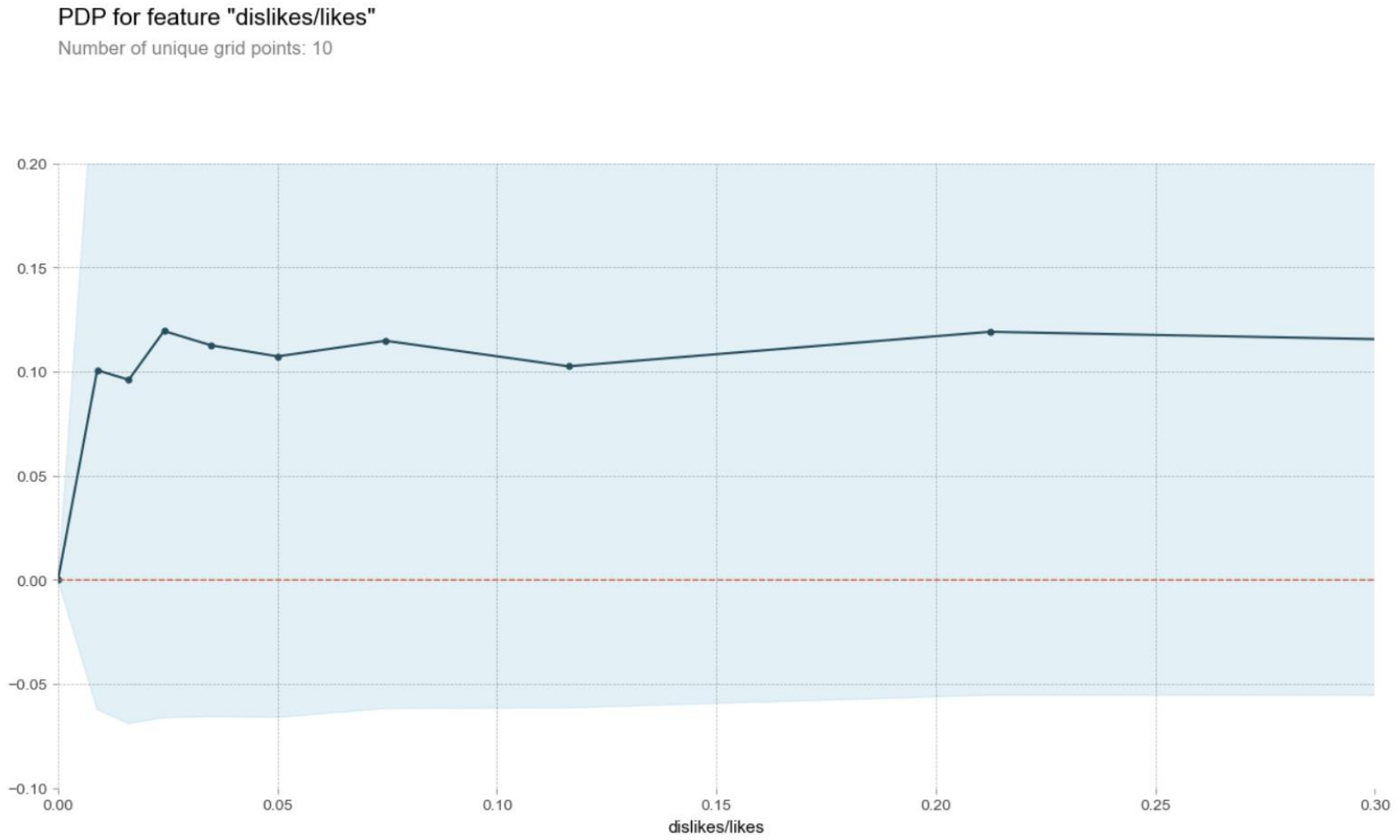
제목의 길이가
길어질수록
target이 1일
확률이 낮아짐



모델 해석: PDP PLOT

dislikes/likes

가설과는 달리
오히려 확률이
높아지는 양상



모델의 한계 및 개선할 점

1. 모델에 사용된 특성 중 조회수 대비 좋아요의 수, 조회수 대비 댓글 수 등의 정보는 영상의 초기 정보가 아님
- 즉, 모델의 예측은 이러한 특성이 시간이 지남에 따라 비례하게 증가하는 것에 가정함
- 그러나, 이 특성들은 영상이 업로드 된 초반에 알 수 있는 정보이기도 하므로 사용할 가치가 있는 정보라고 판단
- 영상 초반의 데이터를 수집할 수 있다면, 모델을 더욱 개선할 수 있음

2. 모델의 타겟은 해당 데이터 셋을 바탕으로 기준한 것이기에, '인기 동영상'의 여부를 판단하기에 객관적이지 않을 수 있음
- 그러나, trending에 오른 영상의 데이터 셋이니 만큼 인기 동영상의 특징들을 파악하기에 좋은 데이터 셋과 기준이라고 판단

결론 및 활용 방안

가설 1

유튜브 업로드 초반 영상의 정보는
추후 동영상의 인기 여부에 영향을 미치는 특성일 것이다.

-> 태그의 개수, 영상 제목의 길이, 업로드 시간이 미치는 영향을 파악할 수 있었음

-> 조회수 대비 좋아요의 비율은 앞서 언급한 모델의 한계로 더욱 개선해야할 점이긴 하나, 인기 동영상의 여부에 중요한 특성임을 알 수 있었음

결론 및 활용 방안

가설 2

분류 예측 모델을 활용하여 category별 광고 적합성을 파악하고,
광고 송출의 target을 좁혀 보다 효과적인 광고를 기대할 수 있을 것이다.

-> 정밀도 90% 이상인 분류예측모델을 사용하여 인기 동영상을 예측하고,

이에 따른 target을 좁힐 수 있을 것으로 예상됨

결론 및 활용 방안

예시

애견 용품 광고를 어떤 영상에 송출할 지 결정할 때,

우리는 최신 영상들의 리스트의 정보를 불러와 모델에 입력한 후,

모델이 1이라고 예측한 영상 중

해당 광고 category_id에 적합한 영상에 광고를 송출할 수 있습니다.

Thank you for listening!