

다음 시험에서
나는 몇 점을
받을까?

Code States
Section2 Project
AI_14_이미리

분석 목차

1. 문제 정의

2. EDA(데이터 전처리)

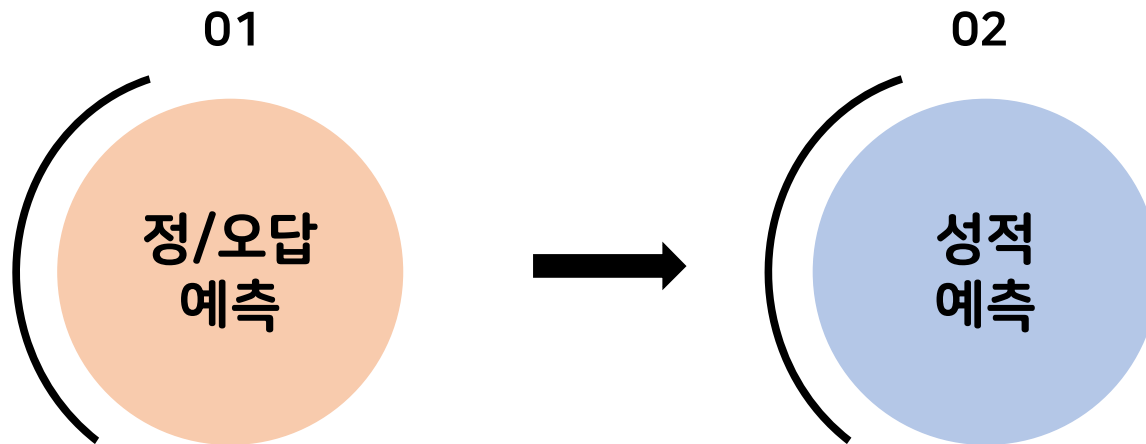
3. 모델링

4. 분석 결과

1. 문제 정의

1. 문제 정의

가설 : 수강생의 과거 정오표 데이터를 통해
다음 시험 정/오답을 예측할 수 있다.



-> 취약 유형 파악

-> 성적 향상 가능

2. EDA(데이터 전처리)

2. EDA(데이터 전처리)

1. 문항 IRT

- testID : 시험지ID
- assessmentItemID : 문제 ID
- difficultyLevel : 난이도 (-5~5)
- discriminationLevel : 변별도 (0~ ∞)
- guessLevel : 추측도(0~1)
- Timestamp : 데이터 생성 일자
- knowledgeTag : 지식체계번호

2. 학생 IRT

- learnerID : 학생 ID
- learnerProfile : 학생 정보
- testID : 시험지 ID
- theta : testID 에 대한 응시자의 능력 수준(-5~5)
- realScore : 진점수
- Timestamp : 데이터 생성 일자

3. 문항 정오답표

- learnerID : 학생 ID
- learnerProfile : 학생 정보 (Gender;학년등급;학년)
- testID : 시험지 ID
- assessmentItemID : 문제 ID
- answerCode : 채점결과 (0:틀림, 1:맞음)
- Timestamp : 응시 일자

2. EDA(데이터 전처리)

4. 문항 IRT + 학생 IRT

- learnerID : 학생 ID
- testID : 시험지 ID
- assessmentItemID : 문제 ID
- answerCode : 채점결과 (0:틀림, 1:맞음)
- testID : 시험지ID
- assessmentItemID : 문제 ID
- difficultyLevel : 난이도 (-5~5)
- discriminationLevel : 변별도 ($0 \sim \infty$)
- guessLevel : 추측도(0~1)
- knowledgeTag : 지식체계번호
- theta : testID 에 대한 응시자의 능력 수준(-5~5)
- realScore : 진점수

3. 문항 정오답표

- learnerID : 학생 ID
- learnerProfile : 학생 정보 (Gender;학년등급;학년)
- testID : 시험지 ID
- assessmentItemID : 문제 ID
- answerCode : 채점결과 (0:틀림, 1:맞음)
- Timestamp : 응시 일자

2. EDA(데이터 전처리)

- theta : testID 에 대한 응시자의 능력 수준(-5~5)
- realScore : 진점수
- difficultyLevel : 난이도 (-5~5)
- discriminationLevel : 변별도 ($0 \sim \infty$)
- guessLevel : 추측도(0~1)
- knowledgeTag : 지식체계번호
- gender (female:1, male:0)
- grade (1~9)

- answerCode : 채점결과 (0:틀림, 1:맞음)

Target

3. 모델링

3. 모델링

target = answerCode



```
# 기준모델 mode
major = y_train.mode()[0]
y_pred=[major]*len(y_train)

from sklearn.metrics import accuracy_score
print('검증 정확도: ', accuracy_score(y_train, y_pred))

#기준모델의 검증 정확도 66%
검증 정확도: 0.6613375400720307
```

기준모델 : mode(최빈값)

시간 순서 정렬 후,
최근 데이터 20% test 셋 생성,
남은 데이터의 최근 20% val 셋 생성

검증 정확도 : 0.661

3. 모델링

1. xgboost

```
from xgboost import XGBClassifier
from sklearn.pipeline import make_pipeline

pipe2 = make_pipeline(XGBClassifier(n_jobs=-1,
                                   learning_rate=0.2))
pipe2.fit(X_train, y_train)

y_pred2 = pipe2.predict(X_val)
print('xgboost 검증 정확도: ', accuracy_score(y_val, y_pred2))

xgboost 검증 정확도: 0.8649572565005739
```

검증 정확도 : 0.865

2. RandomForest

```
from sklearn.ensemble import RandomForestClassifier

pipe = make_pipeline(
    RandomForestClassifier(n_jobs=-1, random_state=28, oob_score=True)
)

pipe.fit(X_train, y_train)

y_pred = pipe.predict(X_val)
print('Random Forest 검증 정확도: ', accuracy_score(y_val, y_pred))

Random Forest 검증 정확도: 0.9320135948074564
```

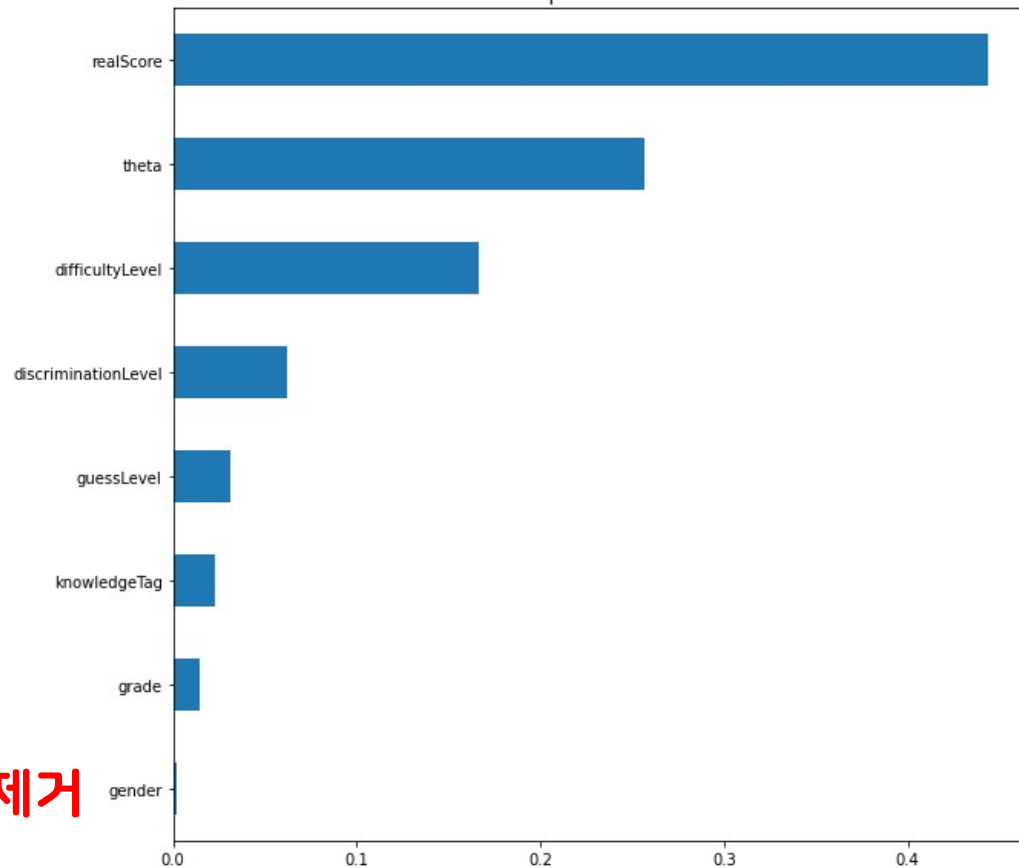
검증 정확도 : 0.932

3. 모델링

특성중요도

1. realScore
2. theta
3. difficultyLevel
4. discriminationLevel
5. guessLevel

gender 제거



3. 모델링

최적의 하이퍼파라미터 찾기

max_depth = 20
max_features = 0.7273
n_estimators = 52

```
from sklearn.model_selection import RandomizedSearchCV
from scipy.stats import randint, uniform

dists = {
    'randomforestclassifier__n_estimators': randint(50, 500),
    'randomforestclassifier__max_depth': [5, 10, 15, 20, None],
    'randomforestclassifier__max_features': uniform(0, 1)
}

clf = RandomizedSearchCV(
    pipe,
    param_distributions=dists,
    n_iter=2,
    cv=2,
    scoring='accuracy',
    verbose=1,
    n_jobs=-1
)

clf.fit(X_train, y_train):
```

```
print('최적 하이퍼파라미터: ', clf.best_params_)
print('MAE: ', -clf.best_score_)
```

```
최적 하이퍼파라미터: {'randomforestclassifier__max_depth': 20, 'randomforestclassifier__max_features': 0.7273335939652188, 'randomforestclassifier__n_estimators': 52}
MAE: -0.9185752414216171
```

3. 모델링

최적의 하이퍼파라미터 검증

```
from sklearn.ensemble import RandomForestClassifier

pipe3 = make_pipeline(
    RandomForestClassifier(n_jobs=-1, random_state=54, oob_score=True, max_depth=20, min_samples_leaf=2, min_samples_split=4, n_estimators=52, max_features=0.7273335939652188)
)

pipe3.fit(X_train, y_train)

y_pred3 = pipe3.predict(X_val)
print('Random Forest 검증 정확도: ', accuracy_score(y_val, y_pred3))

Random Forest 검증 정확도: 0.9231260141686786
```

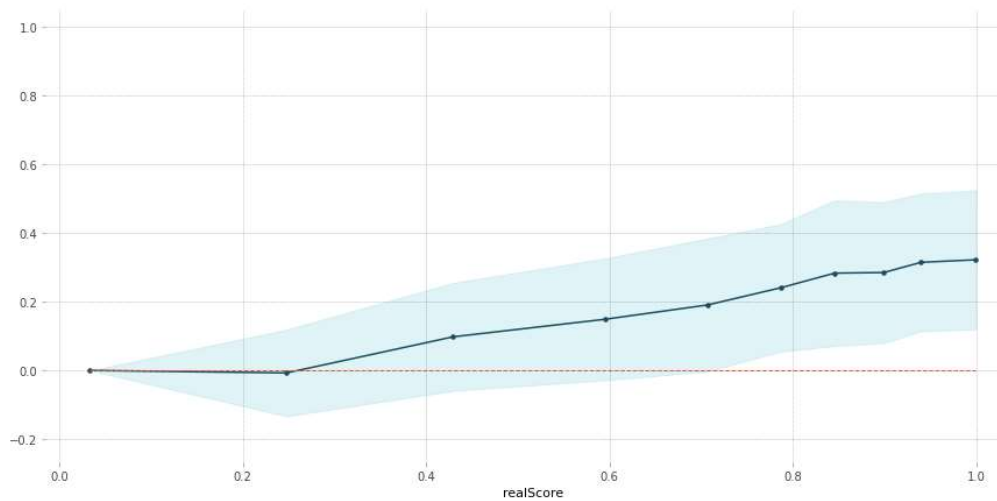
검증 정확도 : 0.923

4. 분석결과

4. 분석 결과 (PDP)

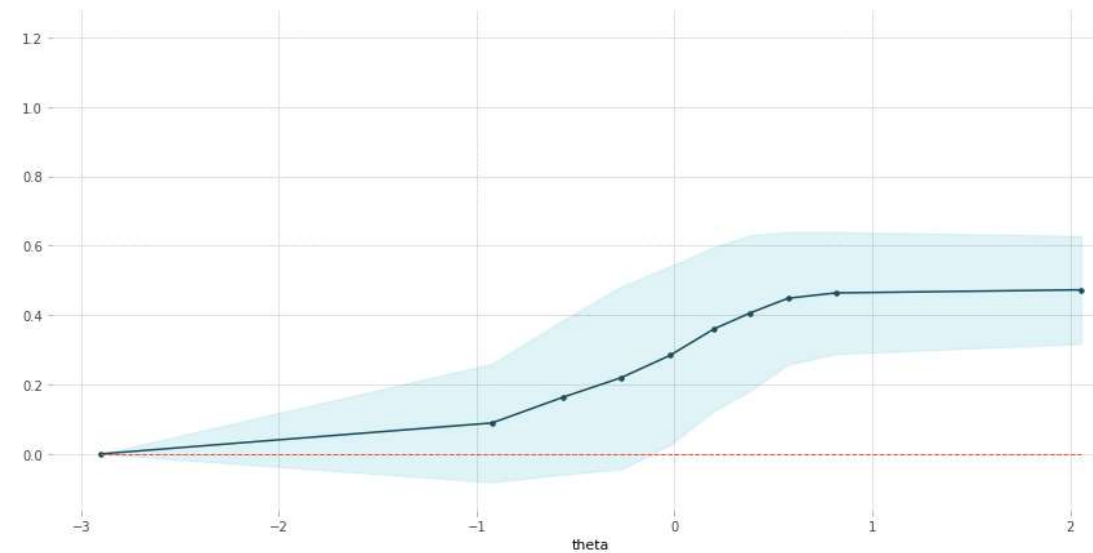
1. realScore

PDP for feature "realScore"
Number of unique grid points: 10



2. theta

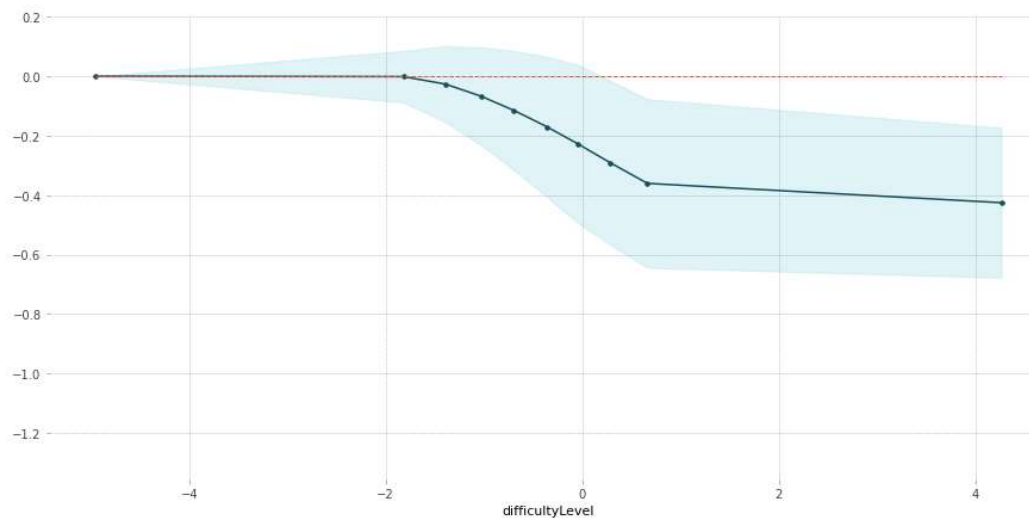
PDP for feature "theta"
Number of unique grid points: 10



4. 분석 결과 (PDP)

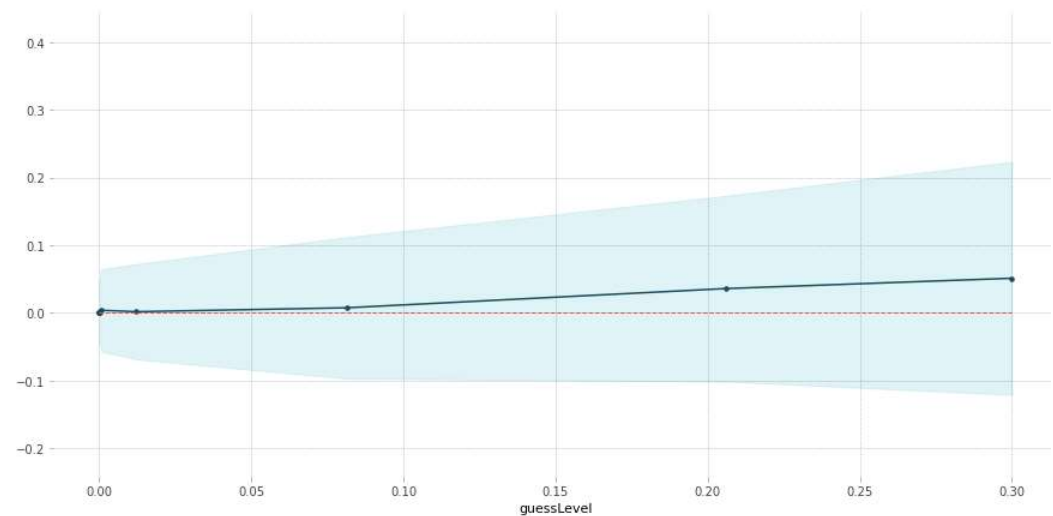
3. difficultyLevel

PDP for feature "difficultyLevel"
Number of unique grid points: 10



4. guessLevel

PDP for feature "guessLevel"
Number of unique grid points: 10



4. 분석 결과

- 1) 문항 정/오답을 **93% 예측** 가능한 모델 생성
 - 2) 진점수, 응시자의 능력 수준이 높을 수록 문항 정답 확률 **UP!**
 - 3) 문제의 난이도가 높을 수록 문항 정답 확률 **DOWN!**
-

더 발전해야 할 점

- 1) **이전 시험의 진점수**를 반영하도록 수정
- 2) 전체가 아닌 개개인의 수강생 ID 를 반영하여 **개인에 대한 분석**
- 3) 분석 환경 및 **효율성 개선**

감사합니다.