

# 3. 학습 플랫폼 구독자 예측 - KNN, optuna

<https://dacon.io/competitions/official/236179/codeshare/9499?page=1&dtype=recent>

## 주제/목표

학습 플랫폼 이용자의 구독 갱신 여부를 예측하는 AI 알고리즘 개발

## 데이터 : 이용자의 플랫폼 이용 정보

- **train**
  - 84406개의 데이터
  - user\_id : 사용자 고유 ID
  - **feature** : 플랫폼 이용 관련 변수
  - **label** : 구독 갱신 여부
    - 0 : 구독 취소 / 1 : 구독 연장
- **test**

## 코드 흐름

### (1) 데이터 전처리

범주형 > 레이블 인코딩 ; **map()** 메서드 사용

- preferred\_difficulty\_level : 선호하는 난이도에 대한 변수, Low / Medium / High  
`train['preferred_difficulty_level'].map({'Low':0, 'Medium':1, 'High':2})`
- subscription\_type : 구독 유형, Basic/ Premium  
`train['subscription_type'].map({'Basic' : 0, 'Premium' : 1})`

### (1-1) Feature Engineering

## (2) Modeling : KNeighborsClassifier()

knn\_objective 함수 생성

### (2-1) best parameter 확인 후 재학습

#### Optuna

Optuna는 **study**와 **trial**을 다음과 같이 정의한다.

- **Study**: objective 함수에 기반하여 optimization을 수행하는 하나의 프로젝트
- **Trial**: Study 내의 optimization 단일 수행

Hyperparameter optimization을 수행하기 위해 **objective**와 **study**를 정의하고, n\_trials 파라미터를 조정하여 몇 회의 trial을 수행할지 설정할 수 있다.

#### Optuna를 통해 하이퍼 파라미터 튜닝하는 과정

##### 1. objective 함수 정의

- **return** 값은 모델의 성능을 표현하는 지표 (mse, f1-macro 등)
- search\_space 설정 : `suggest_*`

```
def knn_objective(trial):  
  
    param_grid = {  
        'n_neighbors': trial.suggest_int('n_neighbors', 1, 50),  
        'weights': trial.suggest_categorical('weights', ['uniform', 'distance']),  
        'metric': trial.suggest_categorical('metric', ['euclidean', 'manhattan']),  
        'p': trial.suggest_int('p', 1, 5)  
    }  
  
    model = KNeighborsClassifier(**param_grid)
```

objective 함수 예시 (수상작 코드)

2. study 객체 생성 `study = optuna.create_study()`
  - direction = minimize / maximize (objective 의 return값에 알맞게 설정)
3. 최적화 수행 `study.optimize(objective, n_trials=100)`
4. 최적의 하이퍼 파라미터 확인 `study.best_trial.params`

### 배운점, 느낀점

optuna 를 통한 하이퍼 파라미터 튜닝 과정을 이해하게 됨

optuna.trial.Trial 클래스의 suggest\_\* 메서드를 이용하여 하이퍼 파라미터를 선택할 수 있다는 것을 배움