인공지능 기초

머신러닝

인공지능_ Day08

2023. 07. 04

CONTENTS

SelectFromModel


```
from sklearn.feature_selection import SelectFromModel

thresholds = model.feature_importances_

for thresh in thresholds:
    selection = SelectFromModel(model, threshold=thresh, prefit=True)
    select_x_train = selection.transform(x_train)
    select_x_test = selection.transform(x_test)
    selection_model = XGBClassifier()
    selection_model.fit(select_x_train, y_train)
    y_predict = selection_model.predict(select_x_test)
    score = accuracy_score(y_test, y_predict)
    print("Thresh=%.3f, n=%d, ACC:%.2f%%"%(thresh, select_x_train.shape[1], score*100))

    #컬럼명 출력
    selected_feature_indices = selection.get_support(indices=True)
    selected_feature_names = [feature_name[i] for i in selected_feature_indices]
    print(selected_feature_names)
```

earlystopping


```
#2. 모델
from xgboost import XGBRegressor
model = XGBRegressor(random_state=123,
                     n_estimators=1000,
                     learning_rate = 0.1,
                     max_depth = 6,
                     gamma= 1)

#3. 훈련
model.fit(x_train, y_train,
          early_stopping_rounds=20,
          eval_set = [(x_train, y_train), (x_test, y_test)],
          eval_metric='rmse')
          # eval_metric 회귀모델 : rmse, mae, rmsle...
          #              이진분류 : error, auc, logloss...
          #              다중분류 : merror, mlogloss...
```

optuna


```
# optuna 적용
import optuna
from optuna import Trial, visualization
from optuna.samplers import TPESampler
from sklearn.metrics import mean_absolute_error
from catboost import CatBoostRegressor
import matplotlib.pyplot as plt
```

optuna


```
def objectiveCAT(trial: Trial, x_train, y_train, x_test):
    param = {
        'n_estimators' : trial.suggest_int('n_estimators', 500, 4000),
        'depth' : trial.suggest_int('depth', 1, 16),
        'fold_permutation_block' : trial.suggest_int('fold_permutation_block', 1, 256),
        'learning_rate' : trial.suggest_float('learning_rate', 0, 1),
        'od_pval' : trial.suggest_float('od_pval', 0, 1),
        'l2_leaf_reg' : trial.suggest_float('l2_leaf_reg', 0, 4),
        'random_state' :trial.suggest_int('random_state', 1, 2000)
    }
    # 학습 모델 생성
    model = CatBoostRegressor(**param)
    CAT_model = model.fit(x_train, y_train, verbose=True) # 학습 진행
    # 모델 성능 확인
    score = r2_score(CAT_model.predict(x_test), y_test)
    return score
```

optuna


```
# MAE가 최소가 되는 방향으로 학습을 진행
# TPESampler : Sampler using TPE (Tree-structured Parzen Estimator) algorithm.
study = optuna.create_study(direction='maximize', sampler=TPESampler())

# n_trials 지정해주지 않으면, 무한 반복
study.optimize(lambda trial : objectiveCAT(trial, x, y, x_test), n_trials = 5)
print('Best trial : score {}, /nparams {}'.format(study.best_trial.value,
                                                  study.best_trial.params))
```

실 습

1. SelectFromModel
2. earlyStopping
3. optuna
4. 팀프로젝트

Day07. 인공지능 Study

1. 인공지능 개념 정리 - 머신러닝, 딥러닝
2. 퍼셉트론 (Perceptron)
3. 다층 퍼셉트론 (Multi-Layer Perceptron: MLP)
4. 옵티마이저 (Optimizer)
5. 학습률 (learning rate)
6. 경사하강법 (Gradient Descent)
7. 손실함수 (Loss Function)
8. 활성화 함수 (Activation Function) - Sigmoid, ReLU, Softmax

9. 회귀분석

10. 결정계수 R2 score

11. 분류분석

12. 원 핫 인코딩 (One Hot Encoding)

13. 난수값 (random_state)

14. 정확도 accuracy score

15. 과적합 (overfitting)

16. 합성곱신경망(CNN)

17. 이미지증강(ImageDataGenerator)

18. 자연어처리(Word Embedding)

19. SVM model

20. Decision Tree model

21. K-Fold

22. Boosting model

23. 그리드서치

24. 배깅

25. 보팅

26. 아웃라이어

수고하셨습니다.