수상작리뷰
[Private 3위, 0.35833] Stacking - DACON

[풍속 예측 AI 해커톤]

◈ 주제

👉 다양한 기상 요소들을 기반으로 풍속 예측 AI 알고리즘 개발

◈ 데이터

- train.csv : 샘플 고유 ID, 월, 일, 측정 시간대(오전/오후/저녁/새벽), 섭씨온도, 절대온도, 이슬점 온도, 상대습도, 대기압, 포화 증기압, 실제 증기압, 증기압 부족량, 수증기 함량, 공기 밀도, 풍향, **풍속**
- test.csv : 샘플 고유 ID, 월, 일, 측정 시간대(오전/오후/저녁/새벽), 섭씨온도, 절대온도, 이슬점 온도, 상대습도, 대기압, 포화 증기압, 실제 증기압, 증기압 부족량, 수증기 함량, 공기 밀도, 풍향
- sample_submission.csv : 샘플 고유 ID , **풍속**

◈ 코드 리뷰

(1) 라이브러리 설치, 모듈 임포트, 데이터 로드

- 커맨드라인 인자 설정

```
import argparse

parser = argparse.ArgumentParser(description="stacking")
parser.add_argument('--best_n', default=4, type=int)
parser.add_argument('--scaler', default="standard", type=str)
parser.add_argument('--cv', default=10, type=int)
parser.add_argument('--seed', default=826, type=int)
args = parser.parse_args('')
```

(2) Feature Engineering

🔷 Extraction

- 계절 피처 ('Season') 추가 (명목형 숫자변수)
- '년중일수' 피처 추가 : (월-1)*30 + 일

🔷 Selection: 월, 일, 측정시간대, 이슬점온도, 대기압, 증기압부족량, 공기밀도, 풍향, Season, 년중일수

🔷 라벨 인코딩: 측정 시간대가 문자형 카테고리 => LabelEncoder.fit()

🔷 표준화

(3) 모델링

- 회귀 모델링: knn, 배깅, ExtraTree, 랜덤포레스트

(4) 스태킹

🔷 함수 정의

```
def get_stacking_ml_datasets(model, X_train_n, y_train_n, X_test_n,
n_folds):

    kf = KFold(n_splits=n_folds, shuffle=True, random_state=seed)

    train_fold_pred = np.zeros((X_train_n.shape[0], 1))
    test_pred = np.zeros((X_test_n.shape[0], n_folds))

    for folder_counter, (train_index, valid_index) in
enumerate(kf.split(X_train_n, y_train_n)):
        X_tr = X_train_n[train_index]
        y_tr = y_train_n[train_index]
        X_te = X_train_n[valid_index]

        model.fit(X_tr, y_tr)
        train_fold_pred[valid_index, :] = model.predict(X_te).reshape(-1,1)
        test_pred[:, folder_counter] = model.predict(X_test_n)

    test_pred_mean = np.mean(test_pred, axis=1).reshape(-1,1)

    return train_fold_pred, test_pred_mean
```

🔷 knn+ bagging+ ets+ rf => LR (메타모델) 학습

🔷 최종 예측

◈ **배울점**

- 상관관계를 살펴본 후, 다중공선성 문제 해결을 위해 적절한 변수를 남긴점.
- 월, 일을 가지고 계절 뿐만 아니라 년중일수라는 피처를 추가할 수 있음을 알게됨.
- 스태킹을 실전에서 어떻게 적용하는지 코드로 확인.