

다양한 모델을 통한 실험

```
from sklearn.ensemble import RandomForestRegressor, GradientBoostingRegressor
from xgboost import XGBRegressor
from lightgbm import LGBMRegressor

# 랜덤 포레스트, GBM, XGBoost, LightGBM model 별로 평가 수행
# n_estimators를 변경할 수 있습니다.
# 구급령을 통해 모델별로 추가 parameter 튜닝하셔도 됩니다.
rf_reg = RandomForestRegressor(n_estimators=1000)
gbm_reg = GradientBoostingRegressor(n_estimators=1000)
xgb_reg = XGBRegressor(n_estimators=500)
lgbm_reg = LGBMRegressor(n_estimators=400)

[68] model = lgbm_reg # 변경 가능

get_model_predict(model, X_train.values, X_test.values, y_train.values, y_test.values, is_exp1=True)

### LGBMRegressor ###
RMSLE: 0.317, RMSE: 47.379, MAE: 29.136

[54] model2 = gbm_reg

get_model_predict(model2, X_train.values, X_test.values, y_train.values, y_test.values, is_exp1=True)

### GradientBoostingRegressor ###
RMSLE: 0.321, RMSE: 50.681, MAE: 31.039
```

10,000개 이상의 학습하기에 충분한 데이터를 갖고 있어 LightGBM과 Gradient Boosting 모델을 선택했습니다. n_estimator는 각각 400, 1000으로 설정했습니다.

두 모델을 혼합하여 계산

여러 모델의 결과를 받고 가중치에 따른 계산을 하는 방법입니다. (단일 모델에서 가장 성능이 좋은 두 모델을 결합하였는데 더 좋지 않은 성능을 보일 수도 있습니다.)

```
[69] # 모델과 학습/테스트 데이터 셋을 입력하면 성능 평가 수치를 반환
def get_two_models_predict(model1, model2, X_train, X_test, y_train, y_test, is_exp1=False):
    model1.fit(X_train, y_train)
    model1_pred = model1.predict(X_test)
    model2.fit(X_train, y_train)
    model2_pred = model2.predict(X_test)

    pred = 0.7*model1_pred + 0.3*model2_pred # 가중치 변경 가능!

    if is_exp1:
        y_test = np.exp1(y_test)
        pred = np.exp1(pred)
    print('### Two Models ###')
    evaluate_regr(y_test, pred)

[70] model1 = lgbm_reg # 변경 가능
model2 = gbm_reg # 변경 가능

get_two_models_predict(model1, model2, X_train.values, X_test.values, y_train.values, y_test.values, is_exp1=True)

### Two Models ###
RMSLE: 0.312, RMSE: 46.930, MAE: 28.852
```

LightGBM에는 0.7의, Gradient Boosting에는 0.3의 가중치를 각각 부여했습니다. 두 모델로 학습한 결과 0.312정도의 RMSLE가 나왔습니다.

