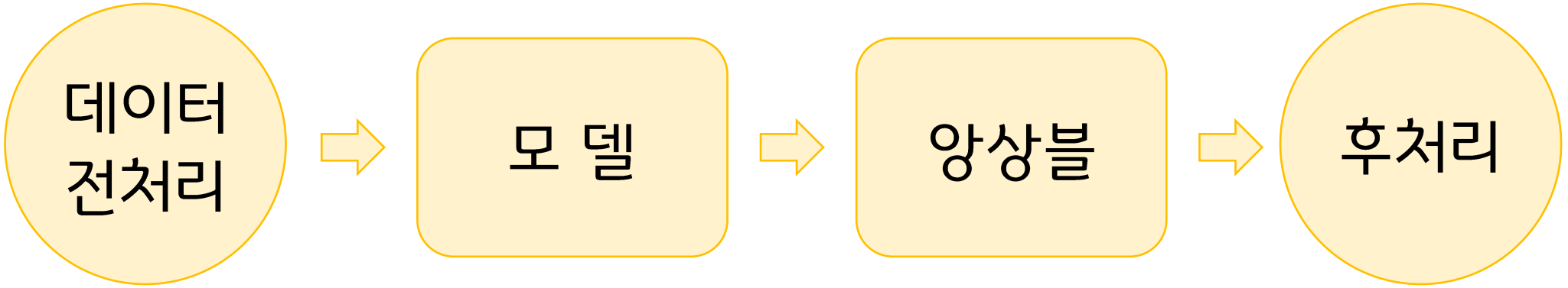
데이콘 감귤 착과량 예측 AI 경진대회

Happy Bus Day

목 차

개 요



- Feature Select
- New Column

- Model
  - Decision tree
  - Random Forest
  - Extra Trees
- Handling
  - GridSearchCV

- Mean Score

- 반올림, 내림

데이터 (1/3) : Feature Select

Feature Select

- correlation을 활용하여 feature select

```
all_names = X_train.columns
selected_mask = selector.get_support()
selected_names = all_names[selected_mask]
unselected_names = all_names[~selected_mask]
print('Selected names: ', selected_names)
```

Feature = 13, 9, 5, 6일 때
NMAE 점수가 가장 높았다.

→ Selected names: Index(['2022-09-01 새순', '2022-09-02 새순',
'2022-09-04 새순', '2022-09-06 새순', '2022-09-07 새순'], dtype='object')

데이터 (2/3) : New Column

New Column from time series data

- Correlation이 높은 데이터가 시계열 데이터인 것에서 착안하여
  전날 데이터와 다음 날 데이터를 이용해 새로운 column을 만들어 학습

```
lst = ['2022-09-01 새순', '2022-09-02 새순', '2022-09-03 새순']]

for i in range( len(lst)-1 ):
    X_train_make['new_column'] =
       (X_train_make[lst[i]] +  X_train_make[lst[i+1]] )/2
```

새로운 column을 생성했을 때
일부 모델의 점수는 상승했고,
일부 모델의 점수는 하락했다.

데이터 (3/3) : Post-process

반올림, 내림

- 데이터 후처리로 모델의 Prediction을 반올림하거나 내림

```
# 반올림
sample_submission['착과량(int)'] =  sample_submission['착과량(int)'].map(lambda x: round(x))

# 내림
sample_submission['착과량(int)'] = sample_submission['착과량(int)'].astype(int)
```

데이터 후처리로 모델의 predictio을 반올림하거나 내림하여 정수 값으로 나타냈다.
이 때 경우에 따라 점수가 상승하였다.

모 델 (1/3) : Model

사용 모델

1. Decision tree
2. Random Forest
3. Extra Trees

선정 기준

평가 지표 NMAE 점수
상위 3개 모델

NMAE : Normalized Mean Absolute Error

```
def NMAE(true, pred):
    score = np.mean(np.abs(true - pred) / true)
    return score
```

모 델 (2/3) : Handling

GridSearchCV

- GridSearchCV를 이용하여 최적의 parameter를 찾고 학습시켜 높은 점수 획득

```
estimator = RandomForestRegressor()

param_grid = {'criterion':['mae','mse'], 'max_depth':[5]}
grid = GridSearchCV(estimator, param_grid=param_grid)
grid.fit(X_train, y_train)

estimator = RandomForestRegressor()
estimator.set_params(**grid.best_params_)

estimator.fit(X_train, y_train)
```

Grid Search를 적용했을 때 트리 계열 regression model의 점수가 상승하였다.

모 델 (3/3) : Ensemble

Mean Score

- Prediction들의 mean값을 활용하여 앙상블

Mean Score를 이용하여 앙상블 했을 때 대부분 점수가 향상되었다.

```
models = [A, B, C, D]

dic = {'ID' :[], '착과량(int)' :[]}

for i in range(len(R['ID'])) :
    tmp = []
    dic['ID'].append(R['ID'][i])
    for esb in models:
        tmp.append( esb['착과량(int)'][i] )

    dic['착과량(int)'].append(np.mean(tmp))

result = pd.DataFrame(dic)
```

시도 아이디어 (1/4) : Learning Machine

미사용 LM 모델

1. SVM
2. KNeighbors
3. Gaussian NB
4. Adaboost
5. Gradient Boosting
6. eXtreme Gradient Boosting
7. Huber
8. Theil-Sen
9. Polynomial (n=2)

모델을 학습시켜 predict한 결과
NMAE 점수가 높지 않아
사용하지 않았다.

시도 아이디어 (2/4) : Deep Learning

미사용 Deep Learning 모델

DNN

Hyper parameters

- **Layer : 1, 2, 3**
- **Dropout : 0.1, 0.3, 0.5, 0.9**
- **Optimizer : Adam, AdamW**
- **Activation : ReLU, Leaky ReLU**

Deep Learning 모델은 이번 task에서 NMAE 점수가 높지 않았다.

데이터 복잡도와 모델 복잡도의 큰 차이로 인해 모델의 학습 능력이 떨어져 점수가 낮은 것으로 판단하여 사용하지 않았다.

시도 아이디어 (3/4) : Ensemble

미사용 Ensemble 모델

Voting Regressor

- sklearn.ensemble의 VotingRegressor 이용 앙상블

분류 task에서 soft 및 hard voting은 좋은 결과를 얻는다는 것을 알고 있어 시도했지만,
회귀 task에서 voting regressor는 만족스러운 결과를 얻지 못해 사용하지 않았다.

```
from sklearn.ensemble
import VotingRegressor


voting = VotingRegressor(
          [('extra tree', extree),
           ('adabst', adaboost),
           ('gradbst', gradboost),
           ('knnr', knnr),
           ('Decision Tree', decitree),
           ('Random Forest', rndforest)])

pred = voting.fit(X_train, y_train)
                          .predict(test)
```

시도 아이디어 (4/4) : Optuna

미사용 모델핸들링 방법

Optuna

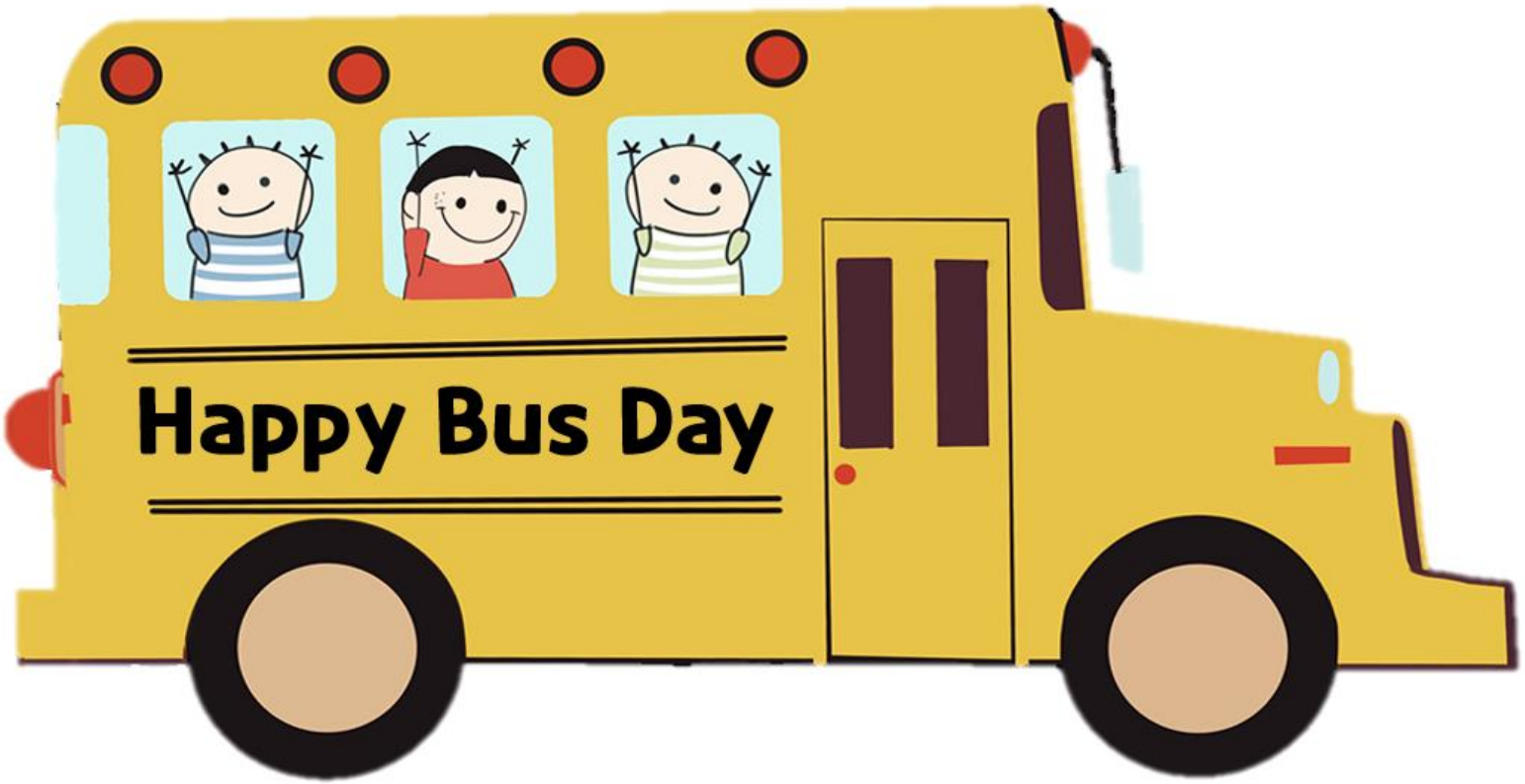
- Optuna를 이용하여 eXtreme Gradient Boosting 모델의 최적 parameter 획득

Optuna를 이용하여 eXtreme Gradient Boosting 모델의 최적 parameter를 찾아 적용했지만 좋은 성능을 얻지 못해 사용하지 않았다.

```
study = optuna.create_study (direction='minimize')
study.optimize(objective, n_trials=50)
```

→ XGBRegressor(learning_rate=0.008,
loss_function='NMAE', max_depth=17,
n_estimators=10000)

```
param = {
    'loss_function': 'NMAE',
    'learning_rate': 0.008,
    'n_estimators': 10000,
    'max_depth': 17
}
xgboost = xgb.XGBRegressor(**param)
```



Thank you for watching

2022.12.14.(수)
Happy Bus Day