HW10

2021321148 INSIK CHO

```
#first classifier
dc = DecisionTreeClassifier(max_depth=3)
CY = y_change(Y)
dc.fit(X,CY)

cm = []
for i in range(len(X)):
  cm.append(dc.predict_proba(X)[i][1])
```

최초 분류는 DecisionTreeClassifier를 사용했고,

```
# start B=101 regression
lr = 0.1 #learning rate
cut_off = 0.5 #cutoff
B=101 # n_ensemble

trees = {}
for i in range(B):
  trees[f'h{i+1}'] = DecisionTreeRegressor(max_depth=3)
  res = CY -cm
  trees[f'h{i+1}'].fit(X,res)
  h = trees[f'h{i+1}'].predict(X)
  cm = cm + lr * h
```

그 다음 101번의 앙상블은 회귀로 진행했습니다.

```
#result

result = []
for i in range(len(tstX)):
  result.append(dc.predict_proba(tstX)[i][1])

for i in range(B):
  result = result + lr * trees[f'h{i+1}'].predict(tstX)

result[result>=0.5] = 1
result[result<0.5] =0

sys.stdout = open(out_name,'w')

print('    ')
print( 'Confusion Matrix( Gradient Boosting)')
print('-------------------------------------------------')
confusion_tst = confusion_matrix(y_change(tstY), result)

accu_tst = 0
for i in range(len(np.unique(y_change(tstY)))):
  accu_tst = accu_tst + confusion_tst[i][i]
accuracy_tst = accu_tst / len(tstdata)

print('           predicted class ₩n Actual 1 ' ,confusion_tst[0], '₩n class  2 ',
for i in range(2, len(np.unique(y_change(tstY)))) :
  print(f'        {i+1} ', confusion_tst[i])
print('model summary')
print('---------------------------------')
print('Overall accuracy = ' ,accuracy_tst)
```

위와 같은 코드로 결과물을 도출한 결과,

```
Confusion Matrix( Gradient Boosting)
-----------------------------------------
          predicted class
Actual 1  [40 47]
class  2  [ 15 124]
model summary
---------------------------------
Overall accuracy =  0.7256637168141593
```

위의 결과를 얻었습니다.