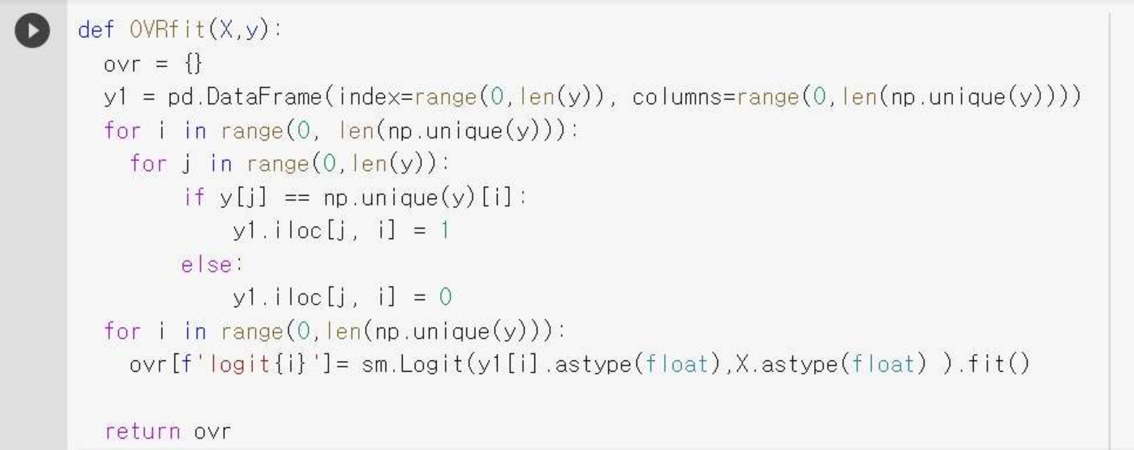
HW5

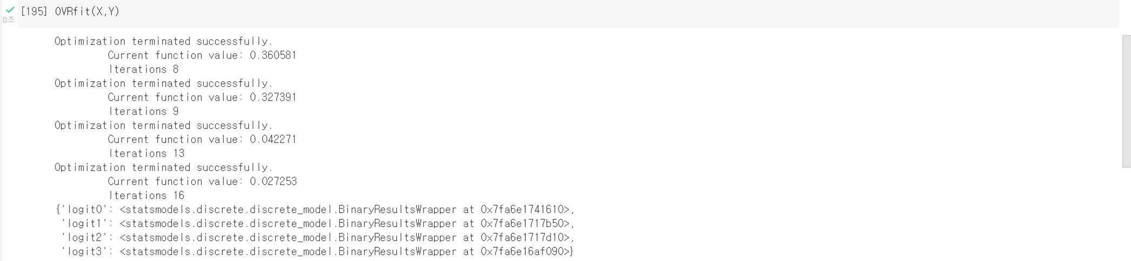
2021321148 INSIK CHO

1.



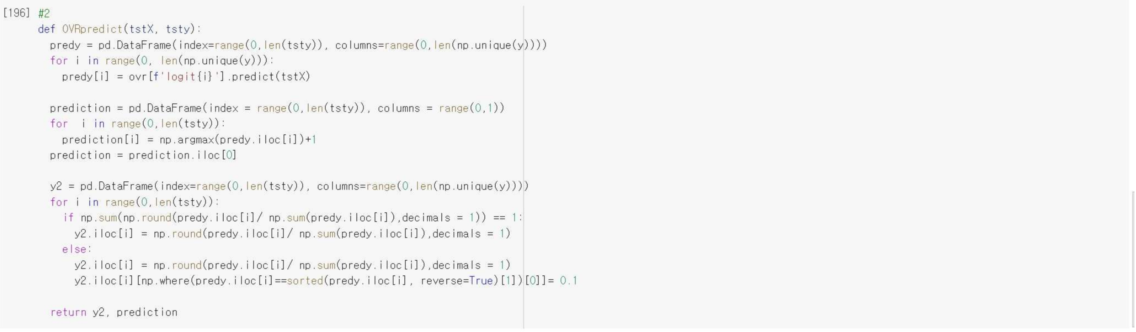
```
def OVRfit(X,y):
  ovr = {}
  y1 = pd.DataFrame(index=range(0,len(y)), columns=range(0,len(np.unique(y))))
  for i in range(0, len(np.unique(y))):
    for j in range(0,len(y)):
        if y[j] == np.unique(y)[i]:
            y1.iloc[j, i] = 1
        else:
            y1.iloc[j, i] = 0
  for i in range(0,len(np.unique(y))):
    ovr[f'logit{i}']= sm.Logit(y1[i].astype(float),X.astype(float) ).fit()

  return ovr
```

위와 같은 코드로 ONE VS RESK classifier를 작성했습니다. (계수 fitting 부분)

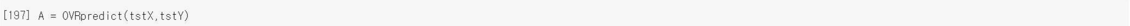


```
[195] OVRfit(X,Y)

      Optimization terminated successfully.
               Current function value: 0.360581
               Iterations 8
      Optimization terminated successfully.
               Current function value: 0.327391
               Iterations 9
      Optimization terminated successfully.
               Current function value: 0.042271
               Iterations 13
      Optimization terminated successfully.
               Current function value: 0.027253
               Iterations 16
      {'logit0': <statsmodels.discrete.discrete_model.BinaryResultsWrapper at 0x7fa6e1741610>,
       'logit1': <statsmodels.discrete.discrete_model.BinaryResultsWrapper at 0x7fa6e1717b50>,
       'logit2': <statsmodels.discrete.discrete_model.BinaryResultsWrapper at 0x7fa6e1717d10>,
       'logit3': <statsmodels.discrete.discrete_model.BinaryResultsWrapper at 0x7fa6e16af090>}
```

해당 코드가 잘 작동함을 확인하였고,

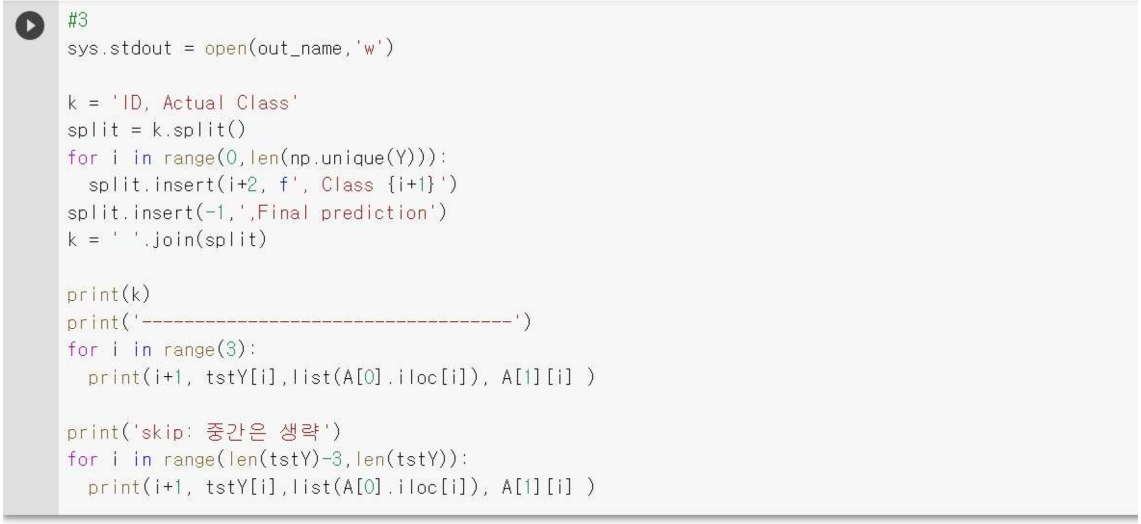


```
[196] #2
      def OVRpredict(tstX, tsty):
        predy = pd.DataFrame(index=range(0,len(tsty)), columns=range(0,len(np.unique(y))))
        for i in range(0, len(np.unique(y))):
          predy[i] = ovr[f'logit{i}'].predict(tstX)

        prediction = pd.DataFrame(index = range(0,len(tsty)), columns = range(0,1))
        for  i in range(0,len(tsty)):
          prediction[i] = np.argmax(predy.iloc[i])+1
        prediction = prediction.iloc[0]

        y2 = pd.DataFrame(index=range(0,len(tsty)), columns=range(0,len(np.unique(y))))
        for i in range(0,len(tsty)):
          if np.sum(np.round(predy.iloc[i]/ np.sum(predy.iloc[i]),decimals = 1)) == 1:
            y2.iloc[i] = np.round(predy.iloc[i]/ np.sum(predy.iloc[i]),decimals = 1)
          else:
            y2.iloc[i] = np.round(predy.iloc[i]/ np.sum(predy.iloc[i]),decimals = 1)
            y2.iloc[i][np.where(predy.iloc[i]==sorted(predy.iloc[i], reverse=True)[1])[0]]= 0.1

        return y2, prediction
```

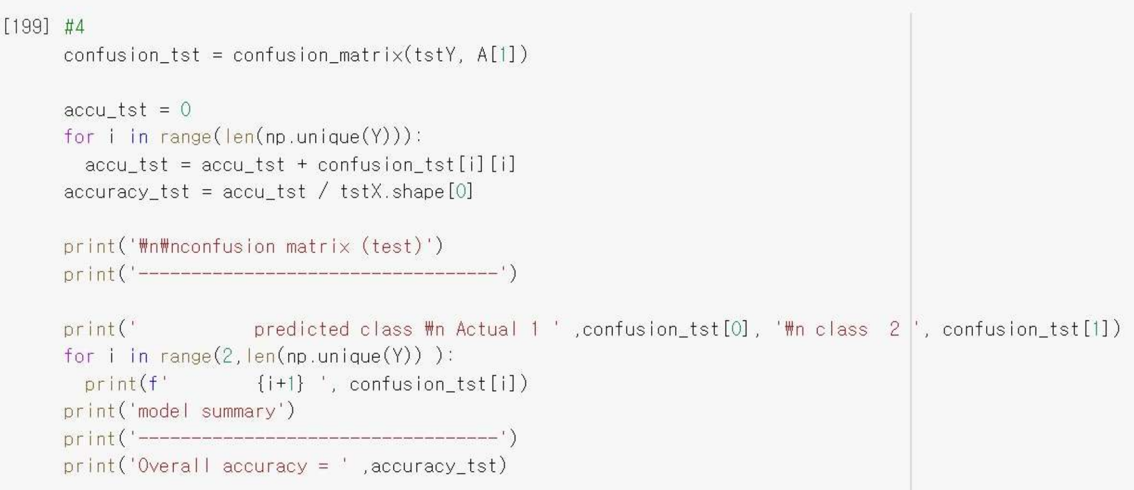


```
[197] A = OVRpredict(tstX,tstY)
```

위와 같은 prediction 코드를 작성한 후 적용했습니다.

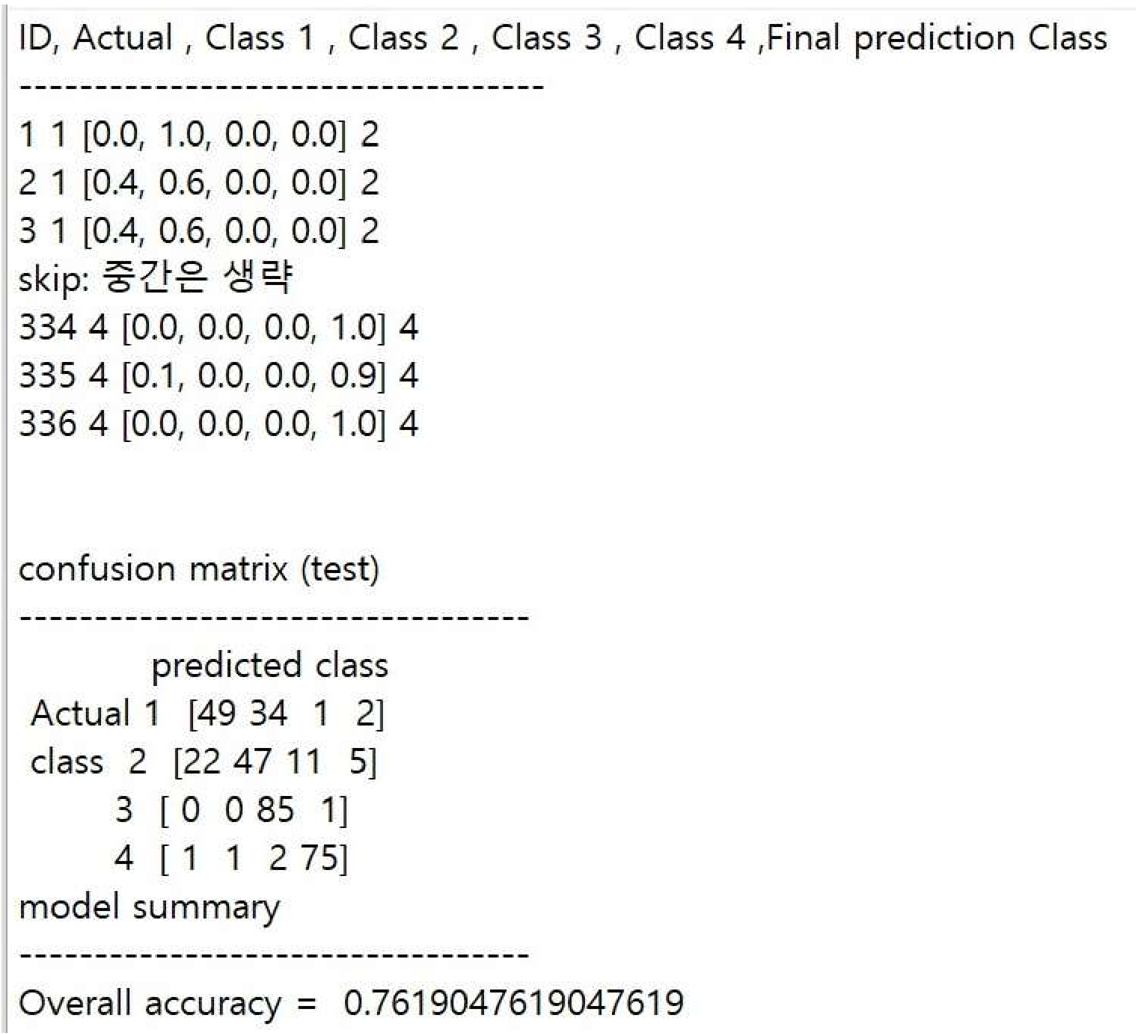


```
#3
sys.stdout = open(out_name,'w')

k = 'ID, Actual Class'
split = k.split()
for i in range(0,len(np.unique(Y))):
  split.insert(i+2, f', Class {i+1}')
split.insert(-1,',Final prediction')
k = ' '.join(split)

print(k)
print('-----------------------------------')
for i in range(3):
  print(i+1, tstY[i],list(A[0].iloc[i]), A[1][i] )

print('skip: 중간은 생략')
for i in range(len(tstY)-3,len(tstY)):
  print(i+1, tstY[i],list(A[0].iloc[i]), A[1][i] )
```



```
[199] #4
confusion_tst = confusion_matrix(tstY, A[1])

accu_tst = 0
for i in range(len(np.unique(Y))):
  accu_tst = accu_tst + confusion_tst[i][i]
accuracy_tst = accu_tst / tstX.shape[0]

print('₩n₩nconfusion matrix (test)')
print('----------------------------------')

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

이어서 과제물에서 요구한 바와 같은 결과물을 도출하기 위한 코드를 작성하고 실행한 결과,



```
ID, Actual , Class 1 , Class 2 , Class 3 , Class 4 ,Final prediction Class
-----------------------------------
1 1 [0.0, 1.0, 0.0, 0.0] 2
2 1 [0.4, 0.6, 0.0, 0.0] 2
3 1 [0.4, 0.6, 0.0, 0.0] 2
skip: 중간은 생략
334 4 [0.0, 0.0, 0.0, 1.0] 4
335 4 [0.1, 0.0, 0.0, 0.9] 4
336 4 [0.0, 0.0, 0.0, 1.0] 4


confusion matrix (test)
----------------------------------
          predicted class
 Actual 1  [49 34  1  2]
 class  2  [22 47 11  5]
       3  [ 0  0 85  1]
       4  [ 1  1  2 75]
model summary
----------------------------------
Overall accuracy =  0.7619047619047619
```

위와 같은 결과물을 얻었습니다.