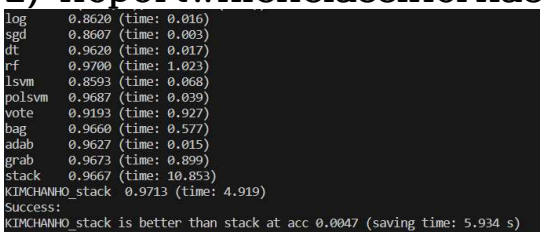
1) Make a stacking classifier better than others

```
# MUST: You should capture the below function in the report!
def get_stacking2(models, nfold=5):
    layer0 = list()
    ### append your chosen models as many as you want
    layer0.append(('rf', models['rf']))
    layer0.append(('df',models['dt']))


    ### You can change layer1 if you are interested in though it is not recommended cuz it's hard work
    layer1 = LogisticRegression()

    model = StackingClassifier(estimators=layer0,
                               final_estimator=layer1,
                               cv=nfold)
    return model

modelstack = dict()
modelstack['KIMCHANHO_stack'] = get_stacking2(models)
```

```
log       0.8620 (time: 0.016)
sgd       0.8607 (time: 0.003)
dt        0.9620 (time: 0.017)
rf        0.9700 (time: 1.023)
lsvm      0.8593 (time: 0.068)
polsvm    0.9687 (time: 0.039)
vote      0.9193 (time: 0.927)
bag       0.9660 (time: 0.577)
adab      0.9627 (time: 0.015)
grab      0.9673 (time: 0.899)
stack     0.9667 (time: 10.853)
KIMCHANHO_stack  0.9713 (time: 4.919)
Success:
KIMCHANHO_stack is better than stack at acc 0.0047 (saving time: 5.934 s)
```

기존의 모델 중 가장 성능이 좋았던 stack(stacking)보다 성능이 0.0047 향상되었으며 수행시간은 5.934s 단축되었음을 알 수 있습니다. 이를 통해 새롭게 만든 모델이 성능과 수행시간 모두에서 기존의 모델보다 우수함을 알 수 있습니다.

2) Reportwhichclassifierhas the highest complexity



위 실험 결과를 통해 가장 시간이 오래 걸리는 모델은 stack(stacking)임을 알 수 있습니다. stacking 방식이 시간이 오래 걸리는 이유는 각층(layer)에서 각각의 모델들을 학습시킨 후 예측을 진행하고 이 예측값을 가지고 다음 층의 모델학습을 진행하기 때문입니다. 그렇기 때문에 n개의 모델을 stacking 했을 때 n개의 모델의 학습시간과 예측시간이 소요되고 이를 bleding한 모델의 학습시간과 예측시간도 추가적으로 소요될 수 밖에 없음으로 시간 복잡도가 가장 높습니다.