1. 代码如下

**from** sklearn **import** datasets  
**from** sklearn.ensemble **import** RandomForestClassifier, ExtraTreesClassifier, GradientBoostingClassifier  
**from** sklearn.model\_selection **import** train\_test\_split  
**from** sklearn.model\_selection **import** StratifiedKFold  
**import** numpy **as** np  
**from** sklearn.metrics **import** roc\_auc\_score  
**from** sklearn.datasets.samples\_generator **import** make\_blobs  
  
**'''创建训练的数据集'''**data, target **= make\_blobs**(n\_samples**=**50000, centers**=**2, random\_state**=**0, cluster\_std**=**0.60)  
  
**'''模型融合中使用到的各个单模型'''**clfs **=** [**RandomForestClassifier**(n\_estimators**=**5, n\_jobs**=-**1, criterion**='gini'**),  
 **RandomForestClassifier**(n\_estimators**=**5, n\_jobs**=-**1, criterion**='entropy'**),  
 **ExtraTreesClassifier**(n\_estimators**=**5, n\_jobs**=-**1, criterion**='gini'**),  
 **ExtraTreesClassifier**(n\_estimators**=**5, n\_jobs**=-**1, criterion**='entropy'**),  
 **GradientBoostingClassifier**(learning\_rate**=**0.05, subsample**=**0.5, max\_depth**=**6, n\_estimators**=**5)]  
  
**'''切分一部分数据作为测试集'''**X, X\_predict, y, y\_predict **= train\_test\_split**(data, target, test\_size**=**0.33, random\_state**=**2017)  
  
  
dataset\_blend\_train **=** np.**zeros**((X.shape[0], len(clfs))) *#33500\*5*dataset\_blend\_test **=** np.**zeros**((X\_predict.shape[0], len(clfs))) *#16500\*5***'''5折stacking'''**n\_folds **=** 5  
skf **= StratifiedKFold**(n\_splits**=**n\_folds)  
*# StratifiedKFold(n\_splits=5, random\_state=None, shuffle=False)*kflods**=**list(skf.**split**(X,y))  
*# skf.get\_n\_splits(X, y)***for** j, clf **in** enumerate(clfs)**:  
 '''依次训练各个单模型'''** print(j, clf)  
 *# 0 RandomForestClassifier(bootstrap=True, class\_weight=None, criterion='gini',  
 # max\_depth=None, max\_features='auto', max\_leaf\_nodes=None,* dataset\_blend\_test\_j **=** np.**zeros**((X\_predict.shape[0], len(kflods)))  
 **for** i, (train, test) **in** enumerate(kflods)**:  
 '''使用第i个部分作为预测，剩余的部分来训练模型，获得其预测的输出作为第i部分的新特征。'''** *# print("Fold", i)* X\_train, y\_train, X\_test, y\_test **=** X[train], y[train], X[test], y[test]  
 clf.**fit**(X\_train, y\_train)  
 y\_submission **=** clf.**predict\_proba**(X\_test)[**:**, 1]  
 dataset\_blend\_train[test, j] **=** y\_submission  
 dataset\_blend\_test\_j[**:**, i] **=** clf.**predict\_proba**(X\_predict)[**:**, 1]  
 **'''对于测试集，直接用这k个模型的预测值均值作为新的特征。'''** dataset\_blend\_test[**:**, j] **=** dataset\_blend\_test\_j.**mean**(1)  
 print(**"val auc Score: %f" % roc\_auc\_score**(y\_predict, dataset\_blend\_test[**:**, j]))  
*# clf = LogisticRegression()*clf **= GradientBoostingClassifier**(learning\_rate**=**0.02, subsample**=**0.5, max\_depth**=**6, n\_estimators**=**30)  
clf.**fit**(dataset\_blend\_train, y)  
y\_submission **=** clf.**predict\_proba**(dataset\_blend\_test)[**:**, 1]  
  
print(**"Linear stretch of predictions to [0,1]"**)  
y\_submission **=** (y\_submission **-** y\_submission.**min**()) **/** (y\_submission.**max**() **-** y\_submission.**min**())  
print(**"blend result"**)  
print(**"val auc Score: %f" %** (**roc\_auc\_score**(y\_predict, y\_submission)))