import numpy as np

import pandas as pd

train\_file = r"\*\\*\\*\train.csv"

test\_file = r"\*\\*\\*\test.csv"

def data\_precess(filename):

df = pd.read\_csv(filename)

data = []

for i in df:

data.append(df[i].astype(np.float32))

if 'CLASS' in df:

label = np.array(df['CLASS'].astype(np.long))

#去掉前面的序号和后面的标签

data = np.array(data)[1:-1].T

else:

label = np.array([])

data = np.array(data)[1:].T

return data,label

train\_data,train\_labels = data\_precess(train\_file)

test\_data,\_ = data\_precess(test\_file)

train\_data = (train\_data+1)/2

test\_data = (test\_data+1)/2

SVC

from sklearn.model\_selection import cross\_val\_score

from sklearn import svm

clf = svm.SVC(C=1, kernel='poly', degree=2, gamma='scale', coef0=0.0, shrinking=True, probability=True,

tol=0.001, cache\_size=200, class\_weight=None, verbose=False, max\_iter=-1, decision\_function\_shape='ovr',

break\_ties=False, random\_state=None)

scores = cross\_val\_score(clf, train\_data, train\_labels, cv=10)

print(scores)

print("Accuracy: %0.2f (+/- %0.2f)" % (scores.mean(), scores.std() \* 2))

from sklearn.model\_selection import cross\_val\_score

from sklearn import tree

clf = tree.DecisionTreeClassifier(criterion='gini', splitter='random', max\_depth=None, min\_samples\_split=7, min\_samples\_leaf=1,

min\_weight\_fraction\_leaf=0.0, max\_features=None, random\_state=None, max\_leaf\_nodes=None,

min\_impurity\_decrease=0.0, min\_impurity\_split=None, class\_weight=None, presort='deprecated', ccp\_alpha=0.0)

scores = cross\_val\_score(clf, train\_data, train\_labels, cv=5)

print(scores)

print("Accuracy: %0.2f (+/- %0.2f)" % (scores.mean(), scores.std() \* 2))

from sklearn.ensemble import RandomForestClassifier

clf = RandomForestClassifier(n\_estimators=1000, criterion='gini', max\_depth=None, min\_samples\_split=2, min\_samples\_leaf=1,

min\_weight\_fraction\_leaf=0.0, max\_features='auto', max\_leaf\_nodes=None, min\_impurity\_decrease=0.0,

min\_impurity\_split=None, bootstrap=True, oob\_score=False, n\_jobs=None,

random\_state=None, verbose=0, warm\_start=False, class\_weight=None, ccp\_alpha=0.0, max\_samples=None)

scores = cross\_val\_score(clf, train\_data, train\_labels, cv=5)

print(scores)

print("Accuracy: %0.2f (+/- %0.2f)" % (scores.mean(), scores.std() \* 2))

AdaBoost

from sklearn.ensemble import AdaBoostClassifier

clf = AdaBoostClassifier(base\_estimator=svm.SVC(C=2, kernel='poly', degree=2, gamma='scale', coef0=0.0, shrinking=True, probability=True,

tol=0.001, cache\_size=200, class\_weight=None, verbose=False, max\_iter=-1, decision\_function\_shape='ovr',

break\_ties=False, random\_state=None),

n\_estimators=1000, learning\_rate=0.7, algorithm='SAMME.R', random\_state=None)

scores = cross\_val\_score(clf, train\_data, train\_labels, cv=5)

print(scores)

print("Accuracy: %0.2f (+/- %0.2f)" % (scores.mean(), scores.std() \* 2))