调包实现随机森林

1. 随机森林与集成算法

集成算法会考虑多个评估器的建模结果，汇总之后得到一个综合的结果，以此来获取比单个模型更好的回归或分类表现。

多个模型集成成为的模型叫做集成评估器，组成集成评估器的每个模型都叫做基评估器。通常来说，有三类集成算法：Bagging、Boosting、Stacking。

随机森林算Bagging的一种，有一点区别是随机森林在构建决策树的时候，会随机选择样本特征中的一部分来进行划分。由于随机森林的二重随机性（数据的随机选取，特征的随机选取），它具有良好的学习性能。

1. 调包实现

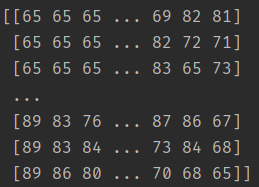
调用sklearn包中的RandomForestClassifier

|  |
| --- |
| from sklearn.ensemble import RandomForestClassifier |

加载数据并修改成使用于包的形式

|  |
| --- |
| def loadData(fileName):  dataArr = []  labelArr = []  fr = open(fileName)  for line in fr.readlines():  curline = line.strip().split(',')  curDataArr = []  for i in range(len(curline[0])):  curDataArr.append(ord(curline[0][i]))  dataArr.append(curDataArr)  labelArr.append(curline[-1])  return dataArr,labelArr  dataArr,targetArr = loadData('newHIV-1\_data/746Data.txt')  dataArr = mat(dataArr)  print(dataArr)  print(targetArr) |

运行：



数据分割成训练集和测试集，测试集占0.3

|  |
| --- |
| Xtrain, Xtest, Ytrain, Ytest = train\_test\_split(dataArr, targetArr, test\_size=0.3) |

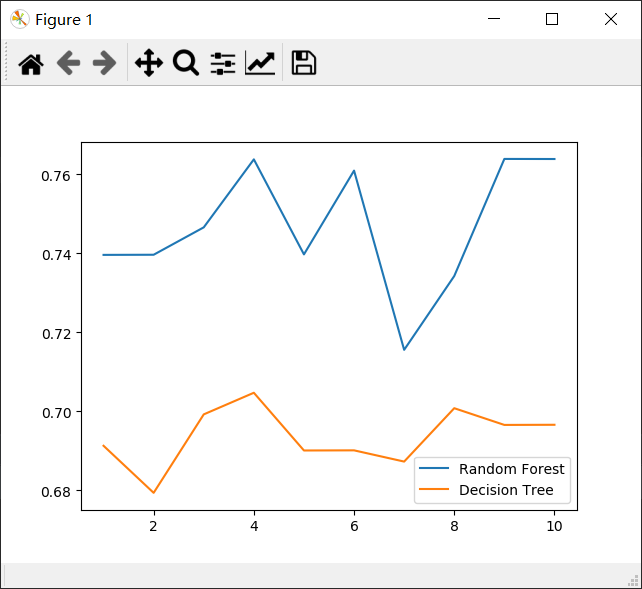
分别训练决策树和随机森林

|  |
| --- |
| clf = DecisionTreeClassifier(random\_state=0)  rfc = RandomForestClassifier(random\_state=0)  clf = clf.fit(Xtrain, Ytrain)  rfc = rfc.fit(Xtrain, Ytrain)  score\_c = clf.score(Xtest, Ytest)  score\_r = rfc.score(Xtest, Ytest) |

运行，得到结果



用图将决策树与随机森林的效果对比（交叉验证）



1. 问题
2. 随机森林的效果确实好于单棵决策树，原因是随机抽取，多次实验，挑选最优？？？
3. 集成学习与多任务学习？？？
4. 关于sklearn中的score函数的计算？？？

分数为预测的系数R²

R² = 1-u/v

u=Σ（真实值-预测值）²

v=Σ（真实值-真实值的平均值）²