通过下图所示代码，我们希望能通过Random Forests即随机森林解决分类问题，这是一个极其经典的通过构建决策树并进行集成算法从而预测分类问题的机器学习或数据挖掘算法。

常用的决策树算法有ID3，C4.5，CART三种。3种算法的模型构建思想都十分类似，只是采用了不同的指标。随机森林是一种重要的基于Bagging的集成学习方法，可以用来做分类、回归等问题。

下面是将决策树与这些算法框架进行结合所得到的新的算法：

1）Bagging + 决策树 = 随机森林

2）AdaBoost + 决策树 = 提升树

3）Gradient Boosting + 决策树 = GBDT

与Bagging集成过程相似，随机森林的构建过程大致如下：

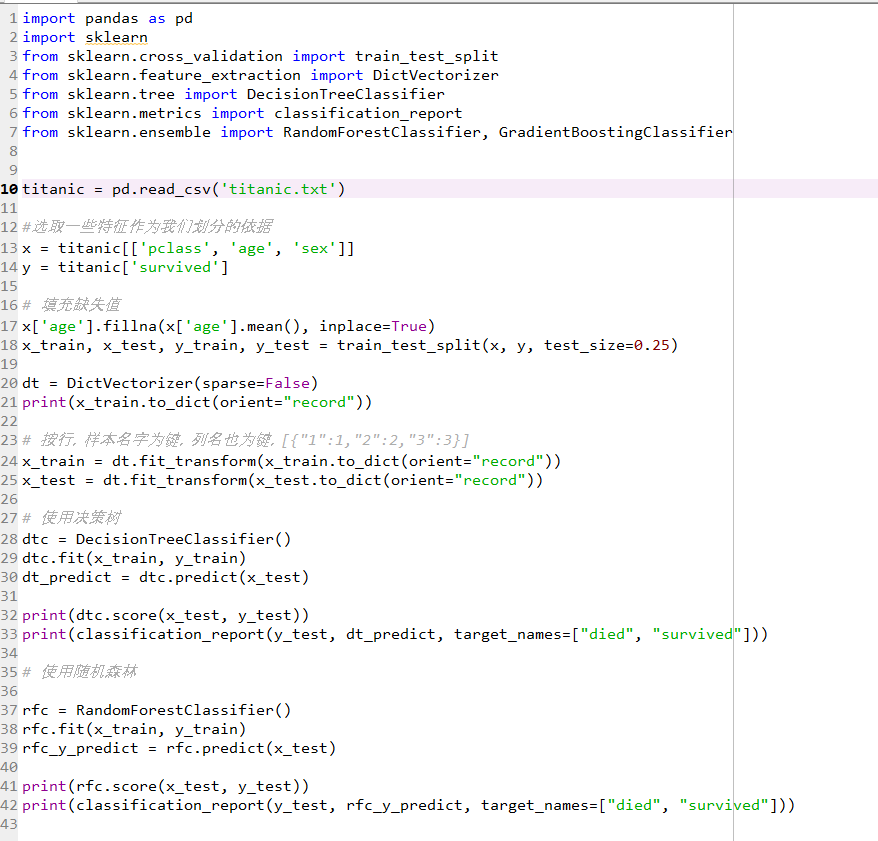
从原始训练集中使用Bootstraping方法随机有放回采样选出m个样本，共进行n\_tree次采样，生成n\_tree个训练集

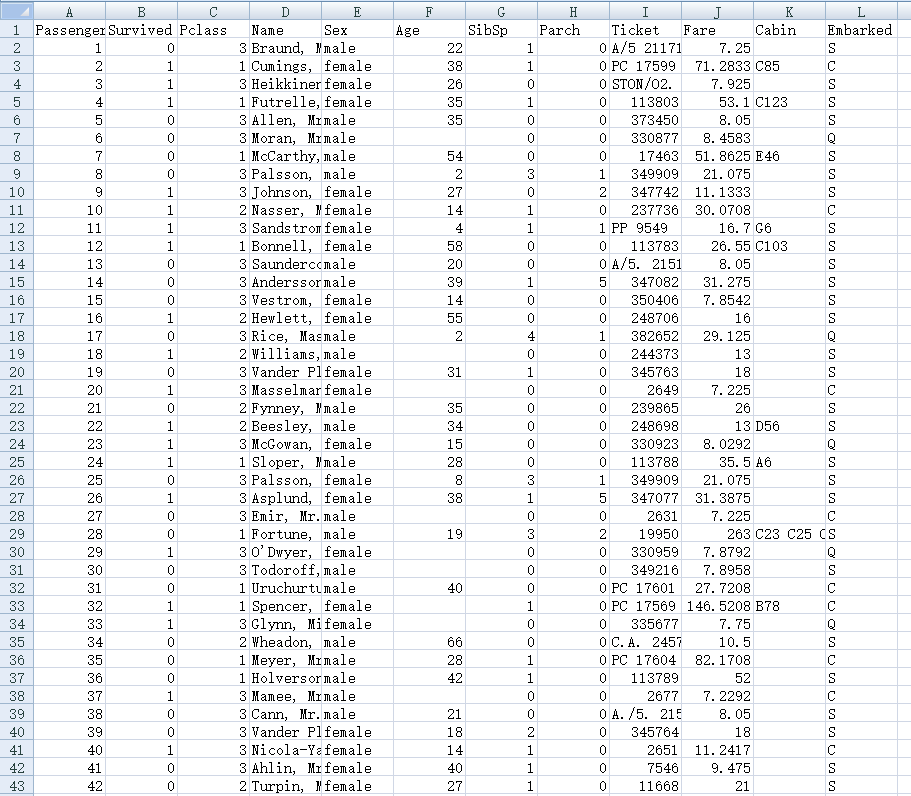
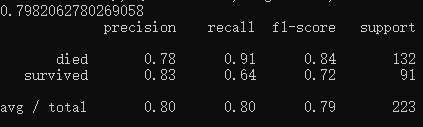
对于n\_tree个训练集，我们分别训练n\_tree个决策树模型

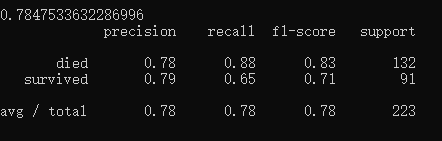
对于单个决策树模型，假设训练样本特征的个数为n，那么每次分裂时根据信息增益/信息增益比/基尼指数选择最好的特征进行分裂

每棵树都一直这样分裂下去，直到该节点的所有训练样例都属于同一类。在决策树的分裂过程中不需要剪枝

将生成的多棵决策树组成随机森林。对于分类问题，按多棵树分类器投票决定最终分类结果；对于回归问题，由多棵树预测值的均值决定最终预测结果。

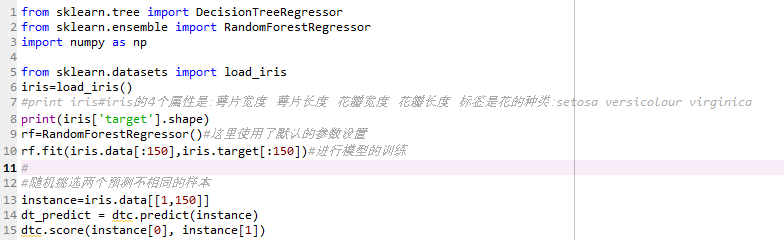


1. 我们这里引用一个极其经典的Titanic生存分析的数据集，通过pd.read\_csv进行调用；
2. 将Titanic提出x与y即预测特征与标签，这里feature我们选择pclass、age、sex，而label为survived；
3. 填充缺失值，例如年龄中的缺失值我们可以按均值填充；
4. 只有train集没有test集，调用cross\_validation.train\_test\_split进行数据集切分，我们设置比例为3：1；
5. 先构建DecisionTreeClassifier决策树分类器，通过fit喂入train，对test计算精度，并打印分类机report；
6. 再构建RandomForestClassifier随机森林分类器，通过fit喂入train，对test计算精度，并打印分类机report；

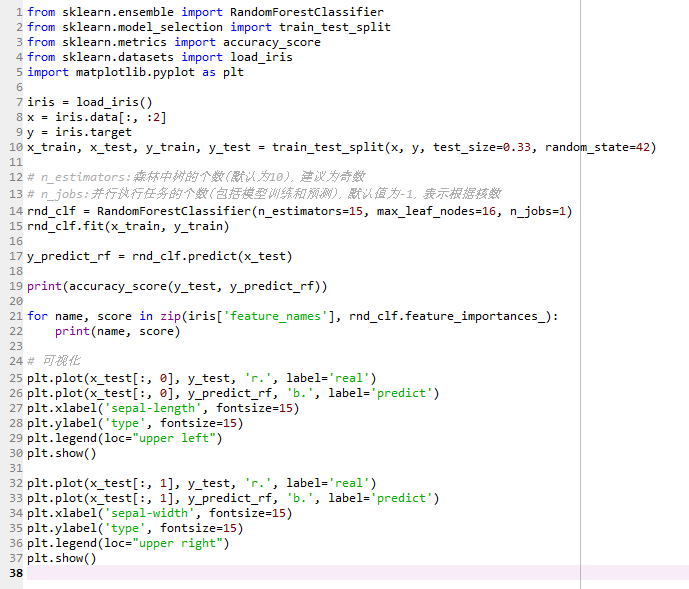


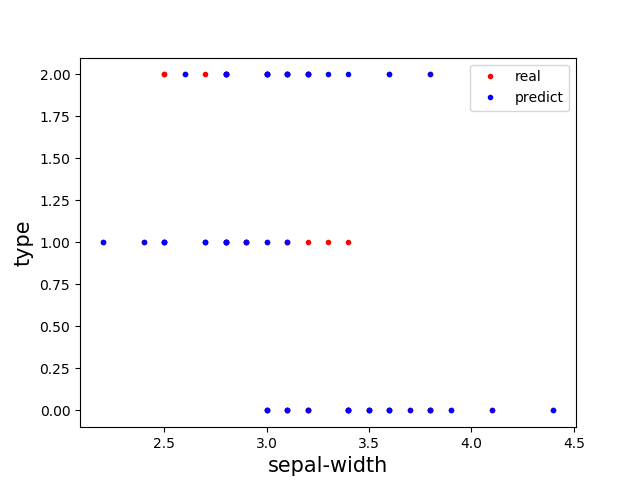
此为我们通过sklearn包中带有的数据集分析，然后我们又找到了两组经典的数据集，然后尝试着解决新环境问题。

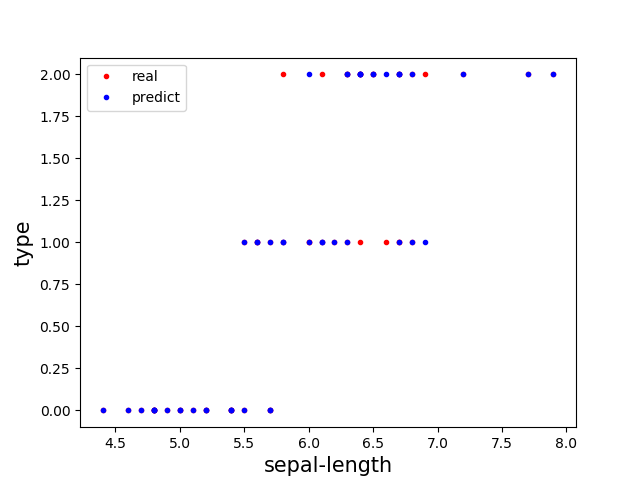
首先我们找到了iris sets，此为经典的鸢尾花问题；其次我们又找到了Titanic生存问题（全部特征）的数据集。同理上述模型，然后我们将新的两组数据进行录入，就得到了我们对这两个新的实际问题的分析从而得到的预测模型。



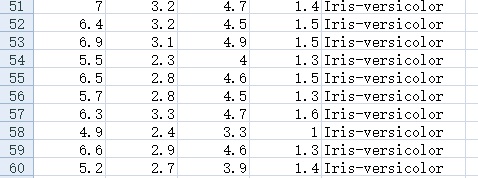
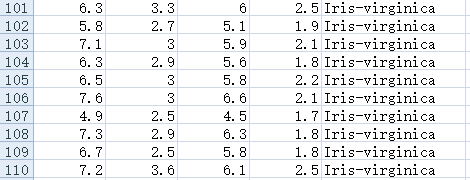
首先是我们写的一个iris的随机森林demo，如下是我们改进并可视化出来的代码。







次为我们通过模型构建，把预测结果可视化出来，根据matplotlib可视化出来的结果我们可以知道，real与predict的结果大致相同，但准确率还可以提高。

这里抽出3中花的类型的每前十条数据，然后录入进模型中即可得到模型。