**实验九 机器学习之分类实验**

**一、实验目的：**

（1）理解KNN算法的数学原理，能够利用Sklearn中KNN分类器解决现实世界中的分类问题；

（2）掌握利用matplotlib对分类结果进行可视化显示的方法，并分析模型的优劣；

（3）掌握利用Sklearn进行分类模型模型评价的方法。

**二、实验仪器及实验环境**

（1）硬件：PC机；

（2）软件：Anaconda Jupyter Notebook，Spyder，Pandas

**三、实验原理**

（1）KNN算法的Sklearn实现

（2）SVM算法的Sklearn实现

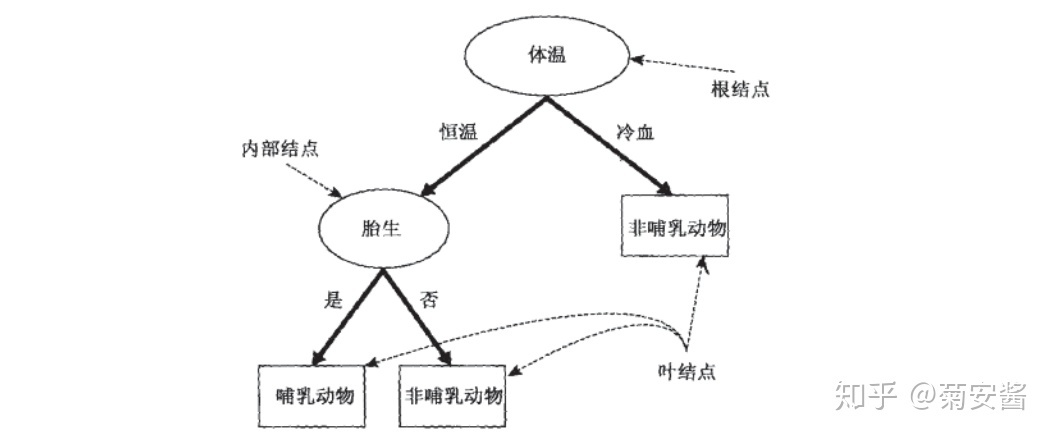
（三）决策树算法的Sklearn实现

决策树是一种原理简单，应用广泛的模型，它可以同时被用于分类和回归问题。决策树（Decision Tree）是一种非参数的有监督学习方法，它能够从一系列有特征和标签的数据中总结出决策规 则，并用树状图的结构来呈现这些规则，以解决分类和回归问题。决策树算法容易理解，适用各种数据，在解决各 种问题时都有良好表现，尤其是以树模型为核心的各种集成算法，在各个行业和领域都有广泛的应用。

决策树算法的本质是一种图结构，只需要问一系列问题就可以对数据进行分类。比如说，来看看下面这组数据集，这是一系列已知物种以及所属类别的数据：



现在的目标是，将动物们分为哺乳类和非哺乳类。那根据已经收集到的数据，决策树算法构造算出了下面的这棵决策树：



sklearn中的决策树算法实现在sklearn.tree模块下这个模块总共包含五个类

tree.DecisionTreeClassiﬁer 分类树

tree.DecisionTreeRegressor 回归树

tree.export\_graphviz 将生成的决策树导出为DOT格式，画图专用

tree.ExtraTreeClassiﬁer 高随机版本的分类树

tree.ExtraTreeRegressor 高随机版本的回归树



代码示例：

from sklearn import tree

trrclassfier=tree.DecisionTreeClassifier()#建立模型对象

trrclassfier.fit(X\_train,y\_train)#通过模型接口训练模型

y\_pred=trrclassfier.predict(X\_test)#通过模型接口对数据预测，提取信息。

print(y\_pred) #查看预测结果

trrclassfier.score(X\_test,y\_test) #利用测试集对模型进行评估

（四）随机森林算法的 Sklearn实现

随机森林算法是集成学习中的一种学习算法。集成学习（ensemble learning）是时下非常流行的机器学习算法，它本身不是一个单独的机器学习算法，而是通过在数据上构建多个模型，集成所有模型的建模结果。基本上所有的机器学习领域都可以看到集成学习的身影，在现实中集成学习也有相当大的作用，它可以用来做市场营销模拟的建模，统计客户来源，保留和流失，也可用来预测疾病的风险和病患者的易感性。在现在的各种算法竞赛中，随机森林、梯度提升树（GBDT）、Xgboost等集成算法的身影也随处可见，可见其效果之好，应用之广。

随机森林是非常具有代表性的Bagging集成算法，它的所有基评估器都是决策树，分类树组成的森林就叫做随机森林分类器，回归树所集成的森林就叫做随机森林回归器。

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。多个模型集成形成的模型叫做集成评估器（ensemble estimator），组成集成评估器的每个模型都叫做基评估器（base estimator）。通常来说，有三类集成算法：装袋法（Bagging），提升法（Boosting）和stacking。

装袋法的核心思想是构建多个相互独立的评估器，然后对其预测进行平均或多数表决原则来决定集成评估器的结果。装袋法的代表模型就是随机森林。装袋法从样本总体中抽取很多个训练集（Training Set），对每个训练集分别拟合模型。将每个模型的结果求平均（Average）。

提升法中，基评估器是相关的，是按顺序一一构建的。其核心思想是结合弱评估器的力量一次次对难以评估的样本进行预测，从而构成一个强评估器。提升法的代表模型有Adaboost和梯度提升树（GBDT）。



Stacking是通过一个元分类器或者元回归器来整合多个分类模型或回归模型的集成学习技术。基础模型利用整个训练集做训练，元模型将基础模型的特征作为特征进行训练。平均法和投票法仅是对弱学习器的结果做简单的逻辑处理，而stacking是再加上一层权重学习器（Meta Learner），基学习器（Base learner）的结果作为该权重学习器的输入，得到最终结果。

（2）随机森林的实现

**from** sklearn.tree **import** RandomForestClassifier *#导入需要的模块*

rfc **=** RandomForestClassifier() *#实例化*rfc **=** rfc**.**fit(X\_train,y\_train) *#用训练集数据训练模型*result **=** rfc**.**score(X\_test,y\_test) *#导入测试集，从接口中调用需要的信息*

重要参数：

（1）控制基评估器的参数

参数 含义

criterion :不纯度的衡量指标，有基尼系数和信息熵两种选择

max\_depth:树的最大深度，超过最大深度的树枝都会被剪掉

min\_samples\_leaf:一个节点在分枝后的每个子节点都必须包含至少min\_samples\_leaf个训练样本，否则分枝就不会发生

min\_samples\_split：一个节点必须要包含至少min\_samples\_split个训练样本，这个节点才允许被分枝，否则分枝就不会发生

max\_features：限制分枝时考虑的特征个数，超过限制个数的特征都会被舍弃，默认值为总特征个数开平方取整

min\_impurity\_decrease：限制信息增益的大小，信息增益小于设定数值的分枝不会发生

（2）n\_estimators

这是森林中树木的数量，即基基评估器的数量。这个参数对随机森林模型的精确性影响是单调的，n\_estimators越大，模型的效果往往越好。但是相应的，任何模型都有决策边界，n\_estimators达到一定的程度之后，随机森林的精确性往往不在上升或开始波动，并且，n\_estimators越大，需要的计算量和内存也越大，训练的时间也会越来越长。对于这个参数，我们是渴望在训练难度和模型效果之间取得平衡。

n\_estimators的默认值在现有版本的sklearn中是10，但是在即将更新的0.22版本中，这个默认值会被修正为100。这个修正显示出了使用者的调参倾向：要更大的n\_estimators。

决策树模型的优点是简单易懂，可视化之后的树人人都能够看懂，可惜随机森林是无法被可视化的。

**四、实验内容：**

（一）基于KNN和随机森林分类算法的森林类型分类

数据集说明：

1-Spruce/Fir：云杉/冷杉

2-Lodgepole Pine：黑松，海滩松

3-Ponderosa Pine：美国黄松

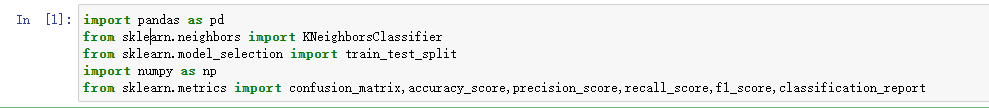
4-Cottonwood/Willow：杨木/杨柳

5-Aspen:(欧洲)山杨;大齿杨

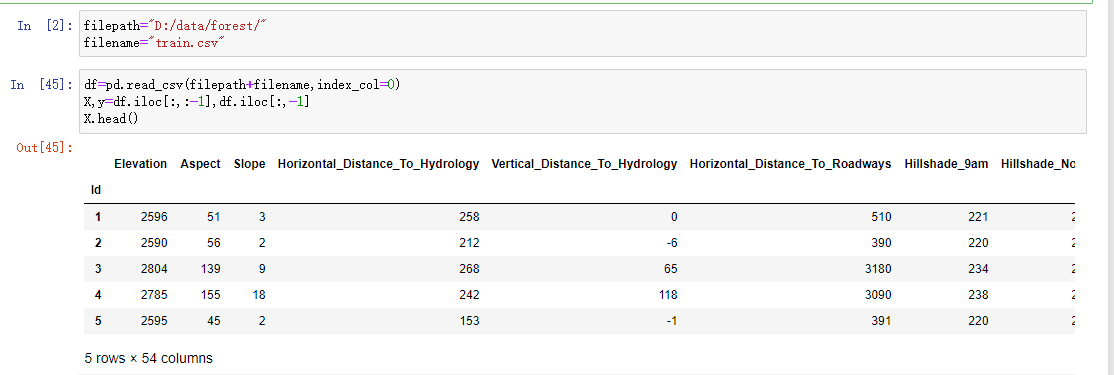
6-Douglas-fir：花旗松,北美西北部的主要树种之一。

7-Krummholz：高山矮曲林，矮盘灌丛

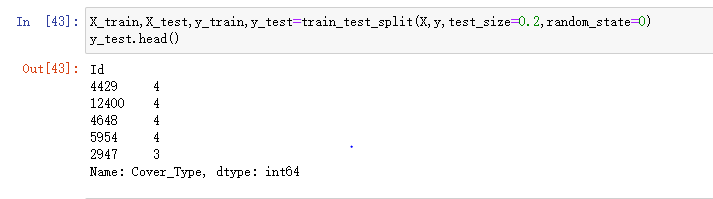
1、导入相关库



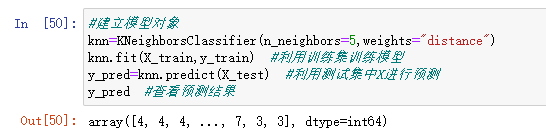
2、打开数据集，分割X和y数据集，并进行数据探索。



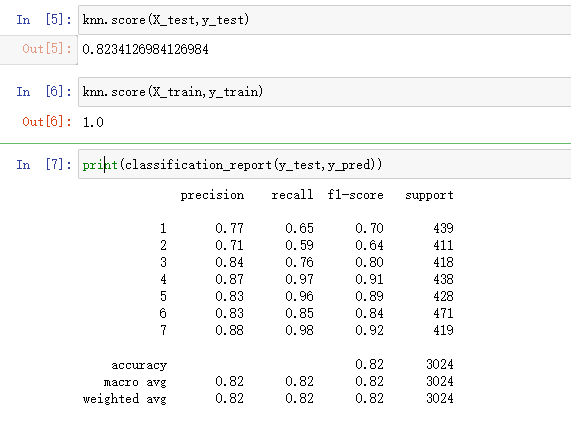
3、分割测试集和训练集



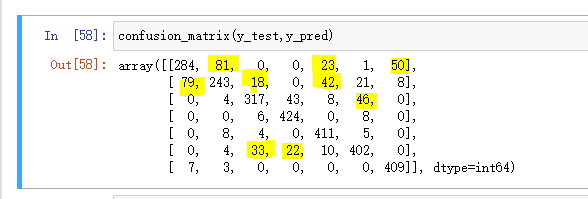
4、利用KNN模型进行森林类型预测。



5、评估模型精度，从中可以看出测试集上k=5时KNN模型的精度为82.3%，从分类报告中可以看出，1和2类型分类精度相对较低，对应云杉/冷杉、Lodgepole Pine和Ponderosa Pine的分类精度相对较低。

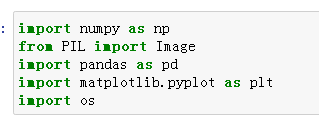


6、显示分类结果的混淆矩阵。从中可以看出1、2之间错分的情况最多，查看数据说明，云杉/冷杉和黑松区分度相对较差。

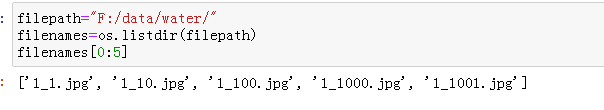


（二）利用水色图像数据集进行水质分类。

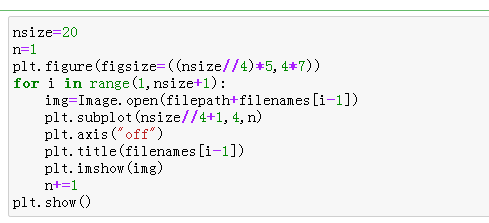
1、导入相关库。

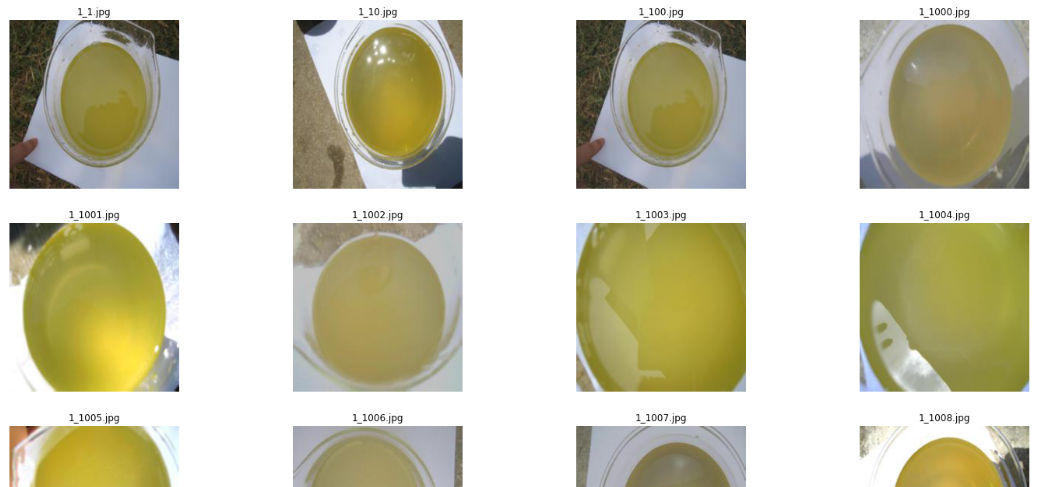


1. 获取图像文件列表

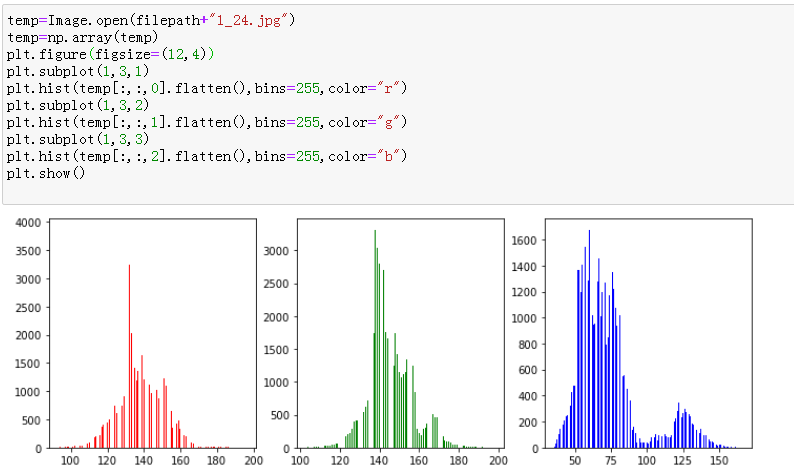


1. 查看部分图像

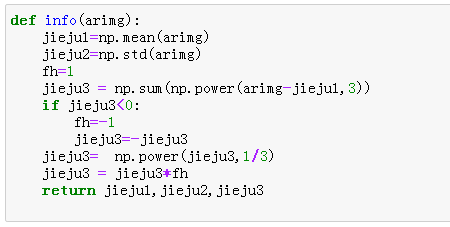


（部分）

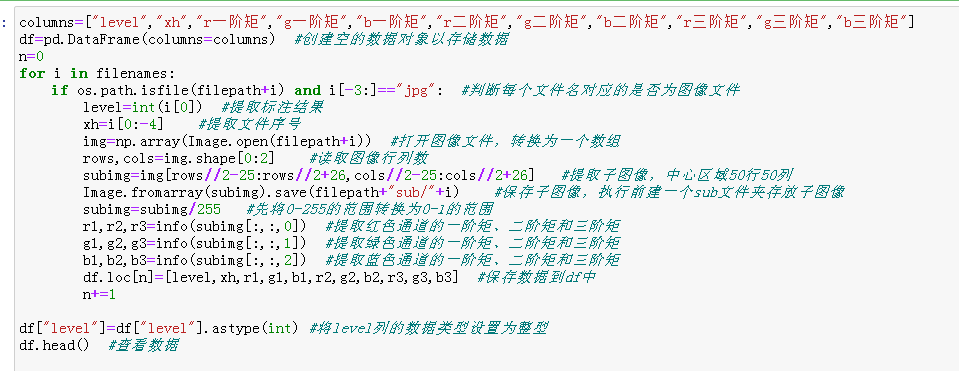
1. 查看某一个图像的形状和直方图。



1. 定义一个函数，求一个波段数据的一阶矩，二阶矩和三阶矩，方便使用。



1. 提取所有图像的信息。

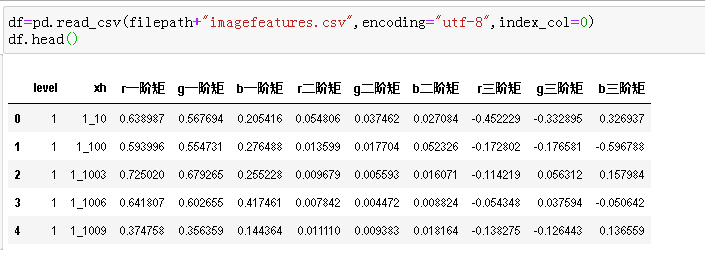




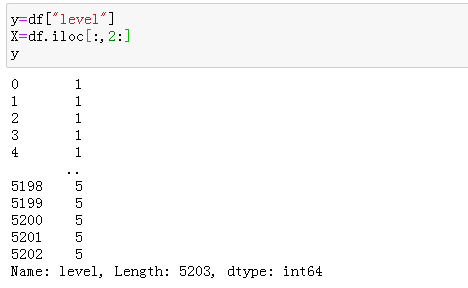
1. 保存df数据到csv文件中，方便下次使用。



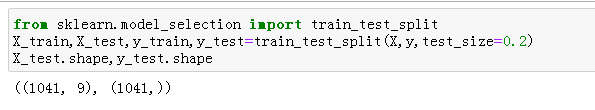
1. 如果实验中途打断，那么之前的图像处理部分将不需要重复进行，只需要打开题7中保存的文件即可。如果没有中途打断，则直接使用上面的df对象即可，跳过本题。



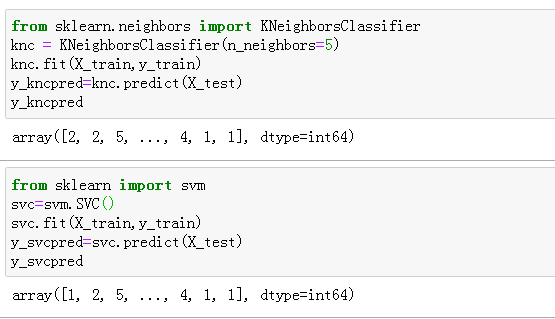
1. 分割X、y数据集。



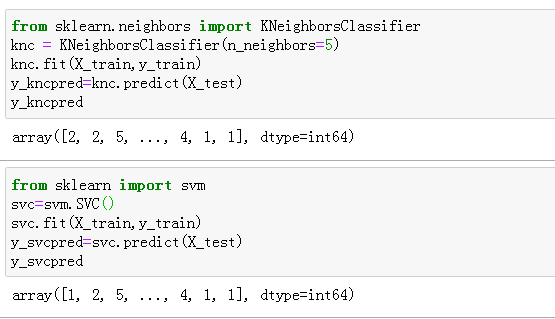
1. 分割训练集和测试集，测试集比例用0.2。



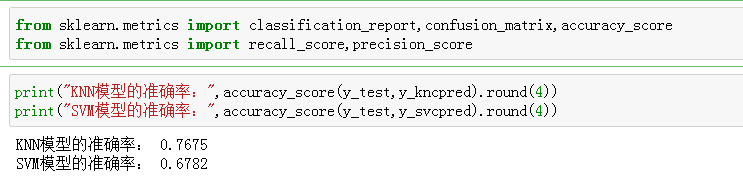
1. 利用KNN建立模型，并利用X测试集预测结果。



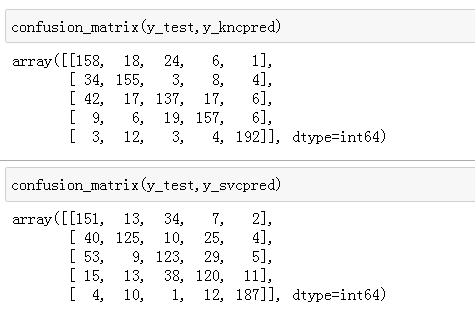
1. 利用SVM建立模型，并利用X测试集预测结果。



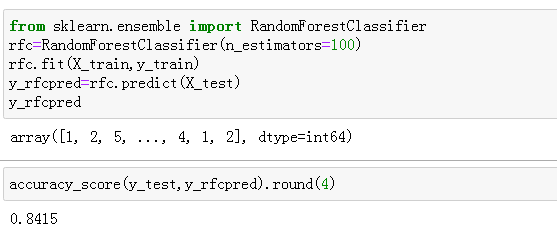
1. 利用测试集评估每个模型的准确率。从中可以看出KNN模型的准确率为76.75%，SVM模型的准确率为67.82%。

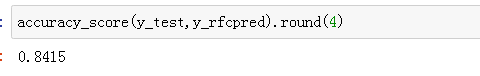


1. 输出两个模型的混淆矩阵，并分析模型的优劣。



1. 可以再选择其他的分类模型，如随机森林来测试模型的优劣。





从中可以看出，随机森林模型的准确率为84.15%，效果较KNN和SVM都要好。当前应该最广泛的分类算法就是随机森林，它对于常规问题的分类效率通常要优于其他分类。

**四、实验结果与分析：**

（1）分析各案例中回归模型的准确性，并从实际角度进行解释。

（2）针对分析过程中出现的问题和错误，写出问题分析和处理过程。