Homework 2

1.利用机器学习算法完成heart-disease数据集的二分类预测

自选两种机器学习算法对heart-disease数据集完成分类预测。数据中包含少量缺失数据（‘?’），同学们可以尝试使用不同的方法来处理缺失值。请同学们自行划分训练集和测试集。需要给出具体实现过程或详细代码、预测准确率、AUC值，并且画出ROC曲线。比较你使用的机器学习算法和分类结果，说说自己的理解。

首先，我决定使用决策树算法对数据进行分类：

源代码见Homework11.py。

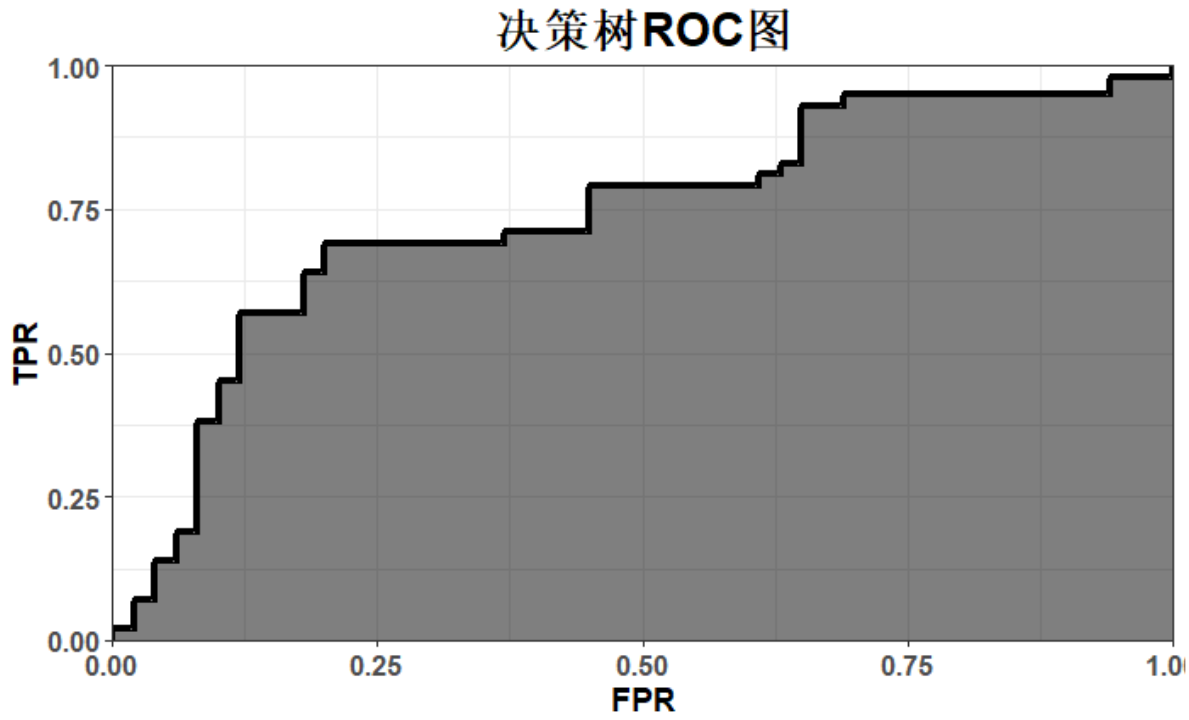
算法中有一些关键点需要阐述：

- 使用了 `KNNImputer类` 用于缺失数据的处理。该类将通过缺失值对应样本的K临近对象的相应值对缺失值进行估算。
- 使用了 `StratifiedShuffleSplit类` 用于进行分层抽样。
- 使用了 `sklearn.tree类` 构建决策树和相应预测
- 之后为进行ROC图的绘制（之后使用到了R进行图形绘制，代码在 `drawing.Rmd` 中），对FPR、TPR值进行了计算，进而最终计算了AUC值。

在一次运行之后，其对应的相关结果分别是：

```
Total Accuracy:0.74
FPR TPR
0.00 0.00
0.00 0.02
0.02 0.02
0.02 0.05
......
0.94 0.98
0.96 0.98
0.98 0.98
1.00 0.98
AUC:0.68
```

根据得到的FPR、TPR值进行作图有：



之后我又使用了KNN算法进行该分类任务。

源代码见Homework12.py。

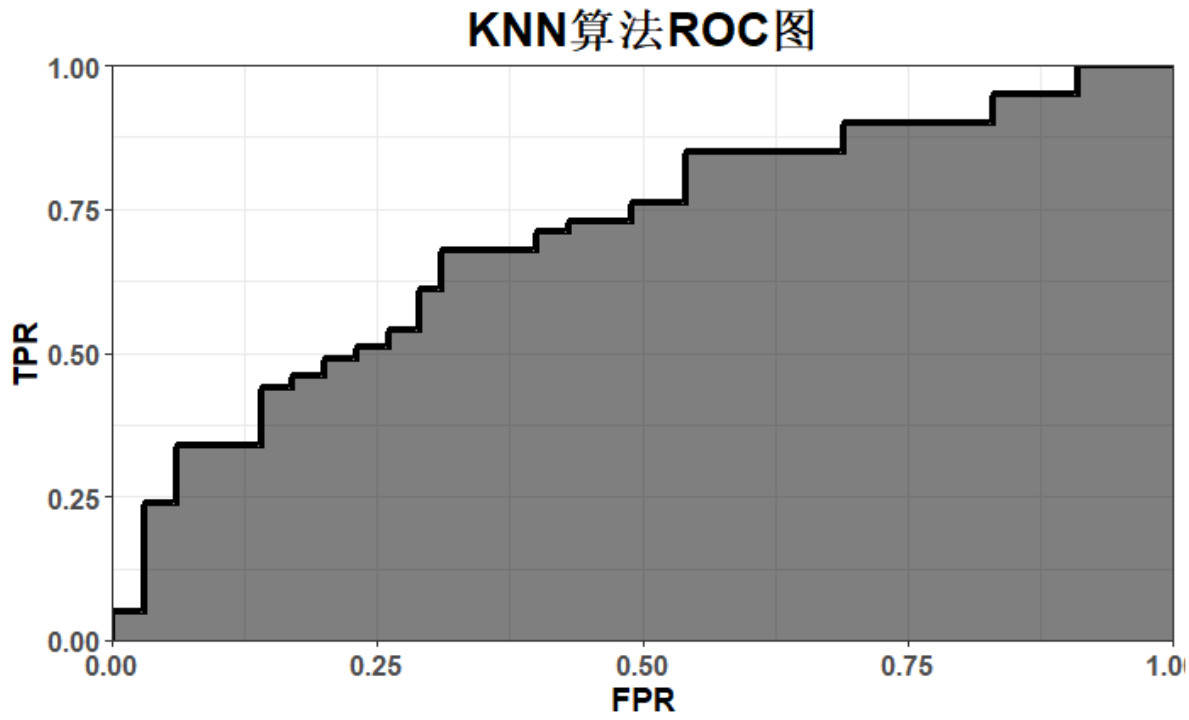
算法中要点包括了：

- 使用了 `SimpleImputer类` 用于缺失数据的处理。该类将依据缺失值对应属性全样本取值及进行缺失值预测。这里以其他样本该属性上的中位数对缺失值进行预测。
- 使用了 `ShuffleSplit类` 用于进行简单随机分组抽样。
- 使用了 `KNeighborsClassifier类` 构建KNN分类同时进行相应预测
- 使用了 `accuracy_score,roc_auc_score,roc_curve` 等函数简化预测精度、AUC值计算
- FPR、TPR值进行两种方式的计算（包括使用 `roc_curve` 函数和自我手动排序循环计算，最终选择了后一种方法）

之后的一次运行结果为：

```
Accuracy:0.65
FPR TPR
0.00 0.00
0.00 0.02
0.00 0.05
0.03 0.05
......
0.91 1.00
0.94 1.00
0.97 1.00
AUC Value:0.70
```

对数据作图有：



总结与比较：由于我选取的算法实际上都很基础（KNN算法和决策树算法），因此能够发现两者的预测准确率实际上是很相似的（同时考虑到每一次训练时决策树算法所含有的随机属性）。但是，根据最终结果能够发现，这两种算法虽然基础，但是实际效果很不错（Accuracy均在0.65且之上），这也符合了决策树算法和KNN算法简单、有效的基本特性。当然，这也说明基础的分类器可能效果瓶颈难以突破，因此需要集成学习等更高级算法进一步提升算法预测效率。

2.利用机器学习算法（K-means和DBSCAN）完成Iris数据集的聚类

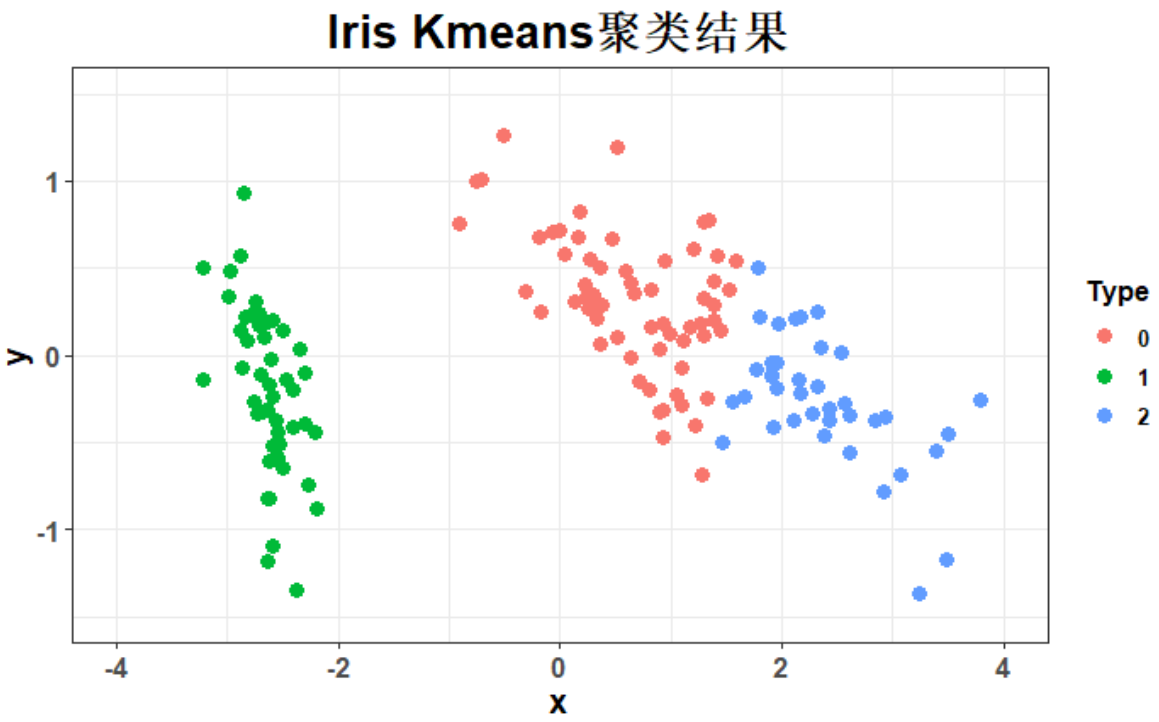
使用两种聚类算法对Iris数据集进行聚类。需要给出具体实现过程或详细代码、聚类结果展示。比较你的聚类结果，说说自己的理解。

首先使用K-means算法，实现代码是：Homework21.py；绘图代码见drawing.Rmd

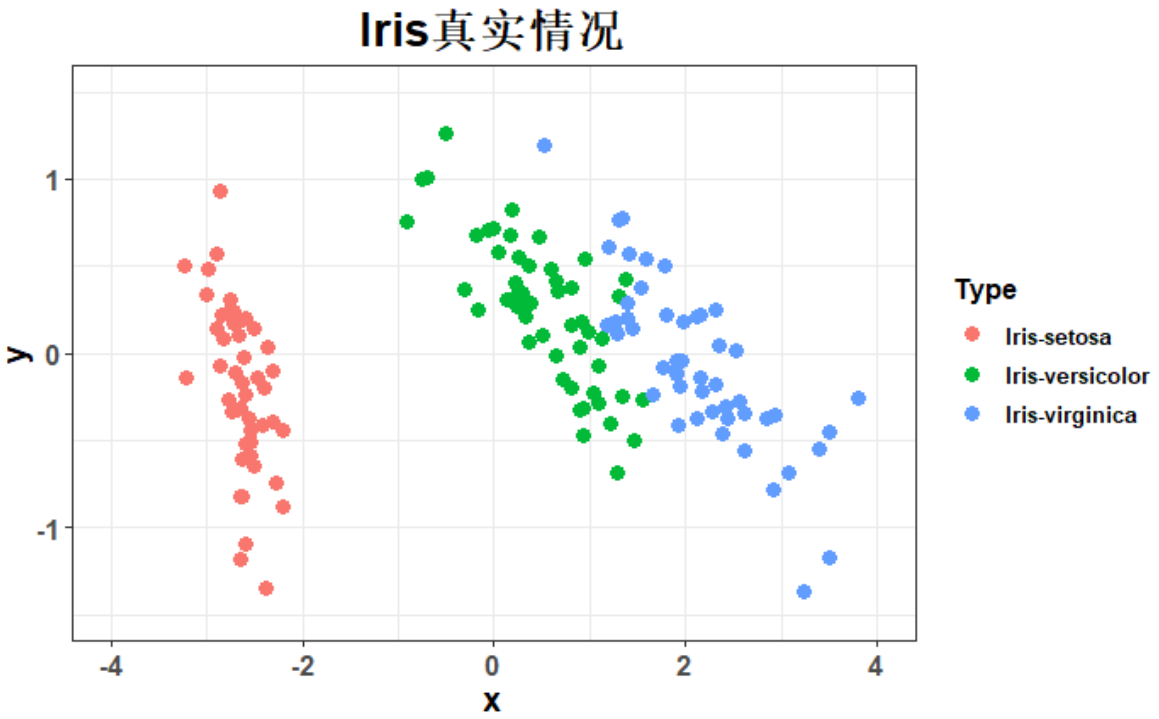
在实现代码中主要的关注点在于：

- 使用了 `KMeans` 类，实现KMeans聚类器，同时制定了聚类器最终结果聚类数为3（考虑到原有数据分为了3类）
- 重新将聚类结果 `classifyer.labels_` 写回到新文件： `iris_data_predict.csv`
- 最终进行结果绘制时，使用了主成分分析的降维方法（否则无法展示四维数据），最终得到X、Y

最终聚类结果为：



（同时列出真实情况）

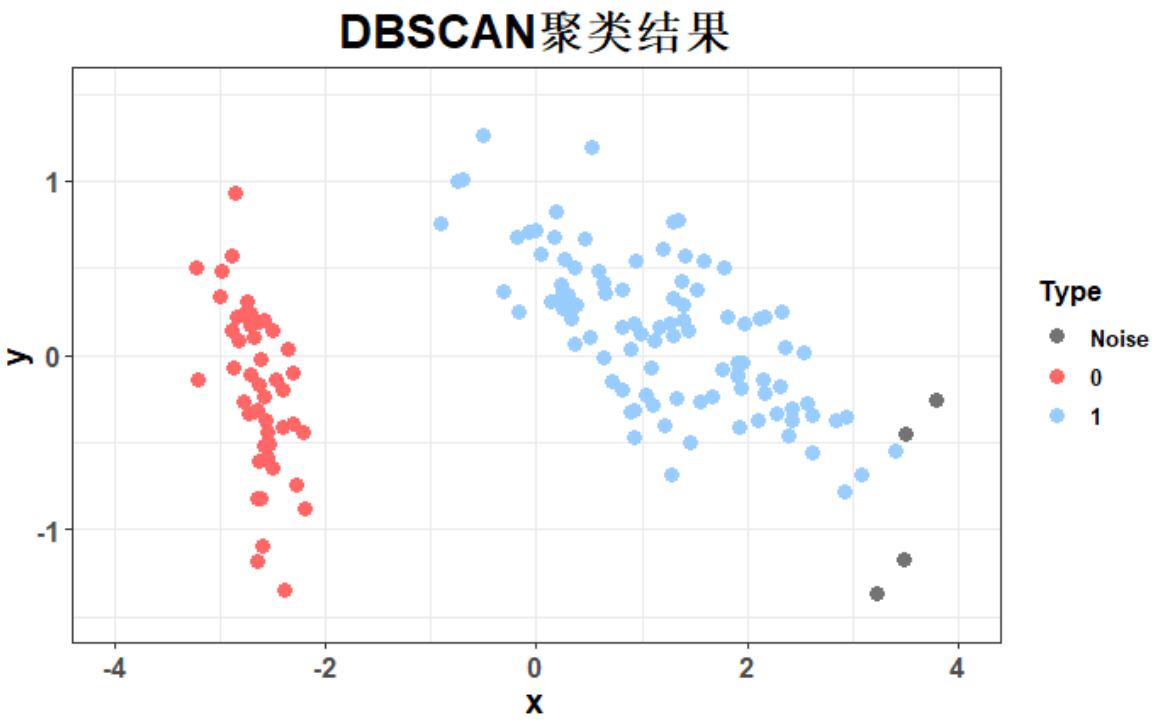


随后进行DBSCAN算法实现，实现文件在Homework22.py中

在实现代码中主要的关注点在于：

- 使用了 `DBSCAN` 类用于实现DBSCAN聚类器
- 为求得适宜的聚类器eps值（聚类中心点到边界点的最大距离），求得了理想分组下样本距离均值，以此作为eps值
- 重新将聚类结果 `classifyer.labels_` 写回到新文件： `iris_data_predict2.csv`
- 最终进行结果绘制时，使用了主成分分析的降维方法（否则无法展示四维数据），最终得到X、Y
- 绘制结果时将未被划分入聚类的的店看作噪声点

最终结果为：



总结：由于本题关于聚类，但给出的数据有实际上存在理想最佳分类标签，似乎较为矛盾，因此仅大致比较聚类结果和实际理想结果（按照分类结果来）。明显发现，DBSCAN表现远劣于K-means（聚类总数错误，此外未能区分两个距离较近的类样本）。而即使是K-means,也能够发现，存在部分样本点聚类出现错误。总的来说，两个算法正确度都不算优秀。我认为原因主要有两个方面：

首先，数据可能未能够取到某些决定性的分类属性，因为我们能够发现 `Iris-versicolor` 和 `Iris-viginica` 两种花朵样本间距离差异很小，甚至两属于不同种的花朵样本间距小于两属于相同种花朵样本的间距，这自然增加了任务难度。

其次，我认为这两个机器学习算法基础而不够复杂，因此难以模拟复杂多变的现实情况。

此外，我认为DBSCAN算法表现特别糟糕的原因是其需要通过估计找到最优的eps值，然而这一过程是十分困难的，因为即使我进行了相关eps值估计操作，仍然无法找到最佳的eps值。