Auto-sklearn源码分析报告

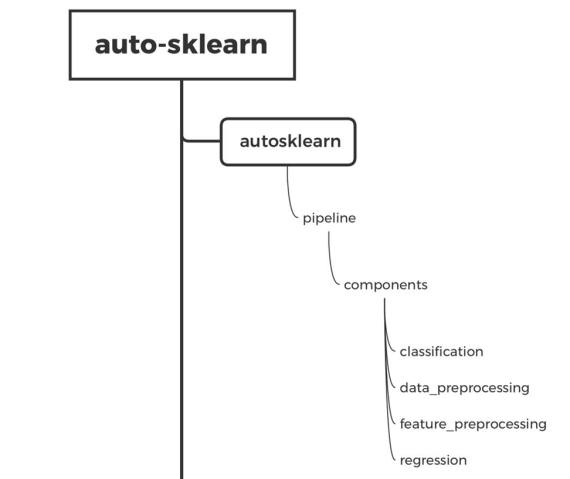
（汇报草稿版）

大家好，我所阅读的源码项目是auto-sklearn，今天我的报告将大概分为以下几个部分为大家介绍auto-sklearn。

首先，什么是auto-sklearn，相信很多人都已经听说过了sklearn，sklearn是基于Python语言的机器学习工具，是机器学习中常用的第三方模块，它对常用的机器学习方法进行了封装，包括回归、降维、分类、聚类等方法。而auto-sklearn，我们从名字可以看出，它就是基于机器学习库sklearn构建的，一个自动化的机器学习工具。

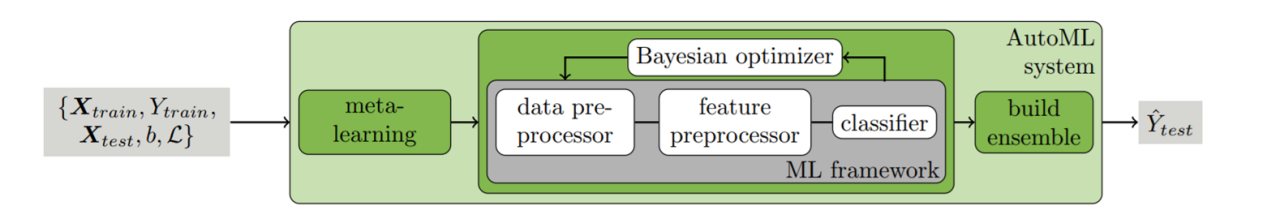


这个就是根据github上的开源代码整理出的它的整体框架，这个看起来可能会有一点乱，不过没有关系，我们这次只需要关注它的精华部分pipeline，pipeline的重点在components。



Components中包括了四个部分，包含16个分类模型，13个回归模型，18个特征预处理方法和5个数据预处理方法，其中不乏大家熟悉的adaboost，决策树，k最近邻算法，随机森林等等，这些都是基于sklearn库实现的，功能非常地强大。

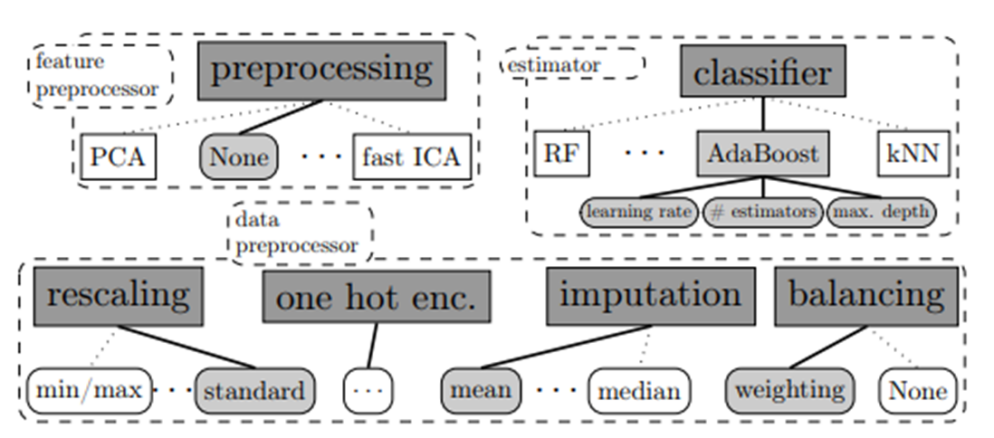
如果仅仅是拥有庞大的模型库和数据处理库，那auto-sklearn和sklearn等其他的机器学习框架也没有什么区别了，所以我们想知道的是这个auto到底可以auto到什么程度呢？



我们以机器学习的分类模型为例，常规的机器学习框架如图中的灰色部分，导入数据后，在经过数据预处理和特征预处理后，通过分类器输出预测值，如果结果不尽人意，需要手动调整超参数并重新选择合适的模型。而自动的部分就如图绿框所示，在ML-framework左边新增meta-learning，在右边新增build-ensemble，并使用贝叶斯优化自动调超参数。meta-learning是用于初始化贝叶斯优化器的元学习，它可以去学习样本数据的模样，一旦找到相似的数据集，就可以根据经验来推荐好用的分类器，比如文本数据用什么模型比较好，很多离散的数据用什么模型比较好。build-ensemble是优化过程中的自动模型集成，可以根据贝叶斯优化找到最佳的分类器组合，往往能提高预测的准确性。

在它的官方论文中提出了一个pipeline的概念很好地总结了auto-sklearn的工作方式，如果把数据想象成自来水，把数据预处理、特征预处理、估计器都想象成水管，模型的表现想象成水的流速，问题其实很简单，那就是**选择什么样的水管**（算法选择），将**水管上的阀门拧到什么样的位置**（超参选择），能让水流的最快。

两个不同的水管也许单独用起来都不咋地，但是拼在一起都奏效了，这就是一个模型集成问题了。

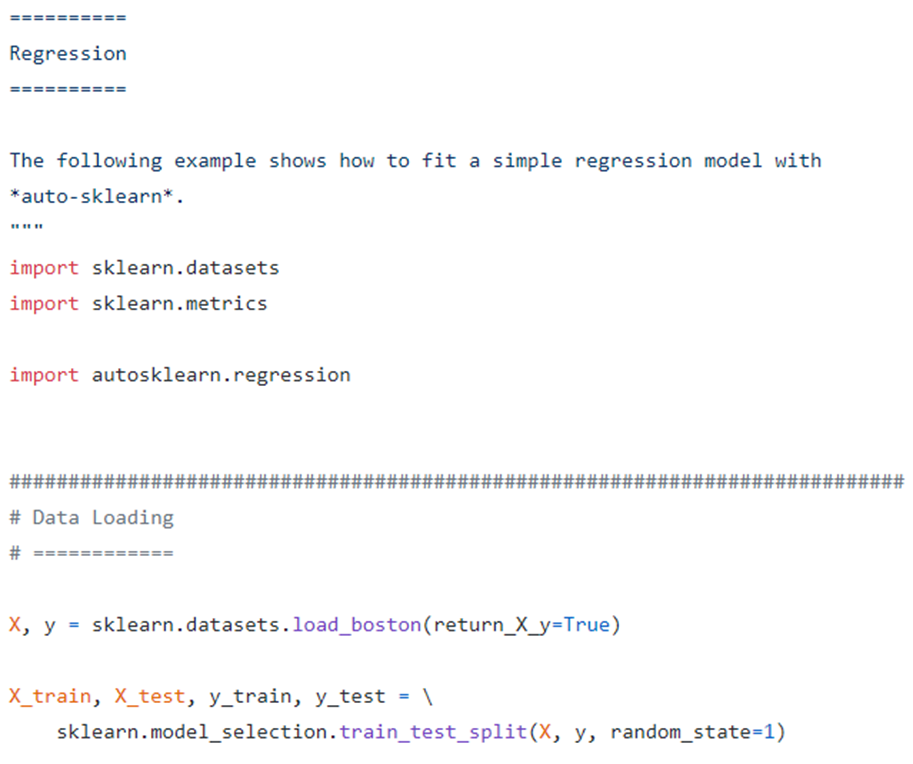


这张图为论文中的图片，我们可以看到，整个过程经过了数据预处理、特征预处理和估计器三根水管，在数据预处理的过程中，数据的balacing我们可以选择做与不做，缺失值填充可以选择用中位数(median)或者平均数(mean)填充，最后的那根管子：估计器(estimator)是一根特殊的管子，他不仅要**选择什么样的水管**（算法选择）,还要知道**将水管上的阀门拧到什么样的位置**（超参选择）。

我们已经对auto-sklearn的工作原理有了一定的了解，那它的适用范围如何呢？虽然我们刚刚看到它拥有非常强大的自动化机器学习的功能，但它的作用域其实是非常有限的。通常我们的机器学习数据有三种，图像数据、文本数据还有表格类的数据，在处理图像数据和文本数据时都有对应的卷积神经网络和自然语言处理的方法，而运用经典机器学习模型进行处理的，只有表格类的数据tabular data，而auto-sklearn的作用域，就是tabular data上的监督学习。

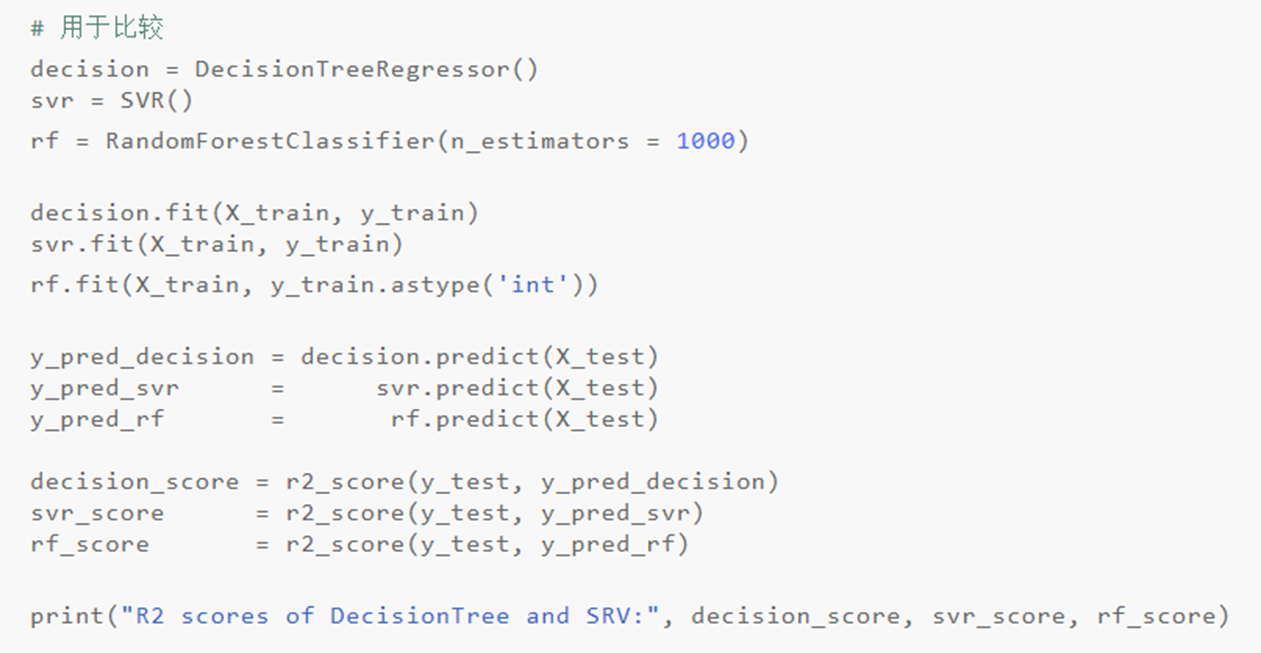
那么对于给定的数据集，我们如何利用auto-sklearn来对它进行自动化的机器学习呢？上手的方法非常简单，设置好以下几种关键的参数即可。第一个是训练时间和内存的使用量的设置，其中时间包括所有模型训练时间的总和以及单个模型训练的最长时间，单位是秒；第二个是训练后模型的存储，参数默认为训练完成后删除训练的暂存目录和输出目录，使用以下参数，可指定其暂存目录及是否删除；第三个是数据的切分，使用resampling\_strategy参数可设置训练集与测试集的切分方法，这里展示了设置五折交叉验证和按比例切分训练集和测试集的方法；最后是模型的选择，可以通过这个参数支持指定备选的机器学习模型，或者从所有模型中去掉一些机器学习模型，如图所示的例子就是将模型指定为只选择随机森机random\_forest。

接下来，我们来看几个官方提供的用例，让我们能够更直观地感受到auto-sklearn的使用方法和功能。



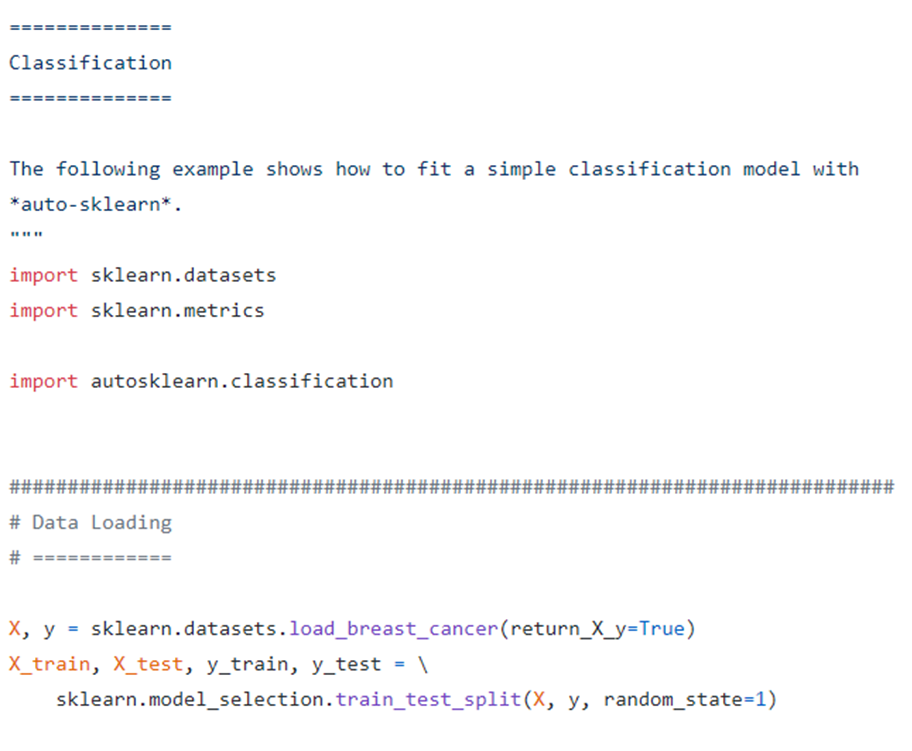


首先是利用auto-sklearn来拟合某个简单的回归模型，导入模块和数据集，为回归模型设置好关键参数后，调用automl中fit函数，开始自动化机器学习，show\_models可以打印出自动集成模型的结果，最后再打印出可以评估回归模型优劣的决定系数R2 score。这里比较了使用auto-sklearn自动搜索模型和其他经典模型如决策树和SVR的R2 score，我们可以发现自动搜索模型的拟合效果是最好的。





接下来展示的是一个在 sklearn 自带的乳腺癌数据集上的自动分类学习，最终打印出来的正确率也是非常高的。





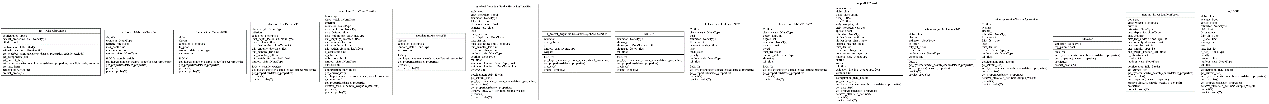
最后，我们来总结一下auto-sklearn这个自动化机器学习框架的优缺点，它的优点有很多，比如，通常情况下，我们只能依据个人的经验，基于机器性能、特征多少、数据量大小、算法以及迭代次数来估计模型训练时间，而auto-sklearn支持设置单次训练时间和总体训练时间，使得工具既能限制训练时间，又能充分利用时间和算力。并且，它支持切分训练/测试集的方式，也支持使用交叉验证，从而减少了训练模型的代码量和程序的复杂程度。另外，它还支持加入扩展模型以及扩展预测处理方法，使训练具有多样性和灵活性。当然它也有一些还可以改进的地方，比如，输出携带的信息较少，想进一步训练只能重写代码，目前适用范围也比较有限，不支持深度学习，对非数值型的数据也不太友好等等。但截至目前，auto-sklearn对自动化机器学习的实现已经超过原本论文写的内容了，相信它会越来越好！

以上就是我报告的全部内容，谢谢大家！

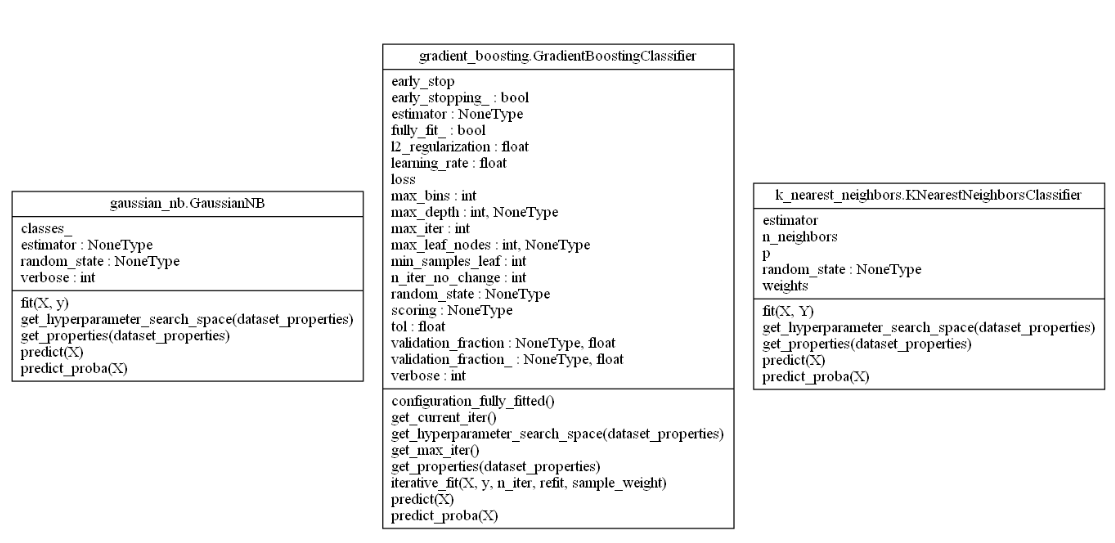
auto-sklearn充分体现了面向对象程序设计中的封装和模块化的思想，其主要功能部分是pipeline中的components，components几乎封装了自动化机器学习所需要的所有功能模块，包括分类器(classification)，回归器(regression)，数据预处理方法(data-preprocessing)，和特征预处理方法(feature-preprocessing)，每一种方法或模块被封装成一个类，其整体组织结构如下：

(图片or结构图)

从文件目录可以清晰地看出，在每一个代表具体功能的文件夹下是各个能实现对应功能的方法或模型，利用graphviz工具在classification目录下生成的类图如下：



可以看到，classification目录下的各个文件实现了17个类，每个类代表一种分类模型，部分类放大如下：



而类与类之间没有任何线条连接，说明各个类之间是相互独立的关系，事实上他们所实现的功能确实是基本相同的，只是实现的方法有所不同。这种将每一种分类方法或模型封装为一个类的设计思路其实也是非常好理解的：将实现功能的具体过程隐藏，保证使用者无法触及不应该触及的部分，避免外部错误导致“交叉感染”，整体上减少程序错误。

