**随机森林**

**C1随机森林-概述**

**当变量的数量非常庞大时，你将采取什么方法来处理数据？**

通常情况下，当问题非常庞杂时，我们需要一群专家而不是一个专家来解决问题。例如Linux，它是一个非常复杂的系统，因此需要成百上千的专家来搭建。

以此类推，我们能否将许多专家的决策力，并结合其在专业领域的专业知识，应用于数据科学呢？现在有一种称为“随机森林”的技术，它就是使用集体决策来改善单一决策产生的随机结果。在这种方法中，每个软件“专家”使用树型算法来完成它们的任务，然后使用这样的树的集合来计算或推导出一个算法模型，最终发现，最终推导出的模型比任何一个专家的输出结果将更优。

决策森林属于集成学习算法，集成学习（ensemble learning）颇受业内关注。集成学习是使用一系列学习器进行学习，并使用某种规则把各个学习结果进行整合从而获得比单个学习器更好的学习效果的一种机器学习方法。集成学习的代表性算法为Boosting系列算法和Bagging系列算法。[[1]](#footnote-1)

**数据决策树**

首先让我们了解基于单个决策树的决策机制。

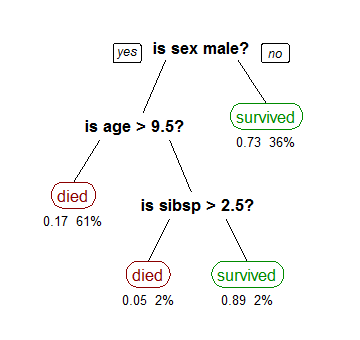
许多大学生同时申请“超级专业技术学院”。入学者的SAT历史平均分数是2,200，GPA的平均分是 4.9。我们获得1,000位申请人的申请信息。我们需要创建一个模型来预测最有可能被录取的学生。我们该怎么做呢？

一种方法是首先将SAT平均分超过2200的申请人划为一组，标注成“更可能”。然后进一步以GPA的分数划分， GPA小于或等于4.9的，划分为“很有可能”，GPA超过4.9的，划分为“非常可能”。

然后，我们以同样的方法处理SAT分数低于2,200的组。GAP分数高的，划分为“可能”，GAP分数低的划分为“不可能”。这看似非常合理，但是存在很多的问题。

* 如果我们先以GPA成绩来划分，然后以SAT成绩划分，那分组结果会一致么?什么才是最好的划分方式呢?
* 如果我们采用更多的标准，例如论文得分、选修课、获奖情况和体育特长等，我们应该用多少特征以及哪些才是最重要的特征呢？
* 在给定平均值的情况下，离散值和异常值如何？对于特征的分布如何影响误差率呢?

决策树利用数据的内部结构进行划分。



维基百科的这个图形[1]表示“泰坦尼克号”幸存者数据集的内在结构。

以上就是一个决策树做决策的框架。 每个节点旁边的数字是生存概率，百分数代表该分类的观测值。每个左分支对应的是“是”，右分支对应的是“否”。 每个绿色节点代表“幸存”，每个红色的节点代表“没有幸存”。配偶或兄弟姐妹的数量记录为“sibsp”。

通过以上可以得知，如果你是男性，并且在较便宜的舱位中，你的生存机会要小得多。 如果你是婴儿或女性，并且在最昂贵的舱位，就会有更大的幸存机会。

**测量决策树“好”的程度**

决策树技术利用信息增益和熵来量化决策树的好坏程度。本质上，我们希望拥有适当的分割点的数量，以便在我们拥有“最佳”的分割方式后，我们不会继续分割群组。那么我们怎么知道一种分割方式比另一种更好呢？

很明显，如果一个分割点非常明确地数据组分割成“接受”和“拒绝”，那么这就是“好”的。

但是，比方说，我们按照论文分数进行了明确的分割，可是每组的GPA分数非常无序，而且STA的分数平均分配。那么，我们实际上没有做出很好的预测，因为两组同样混乱，看不出好坏。

熵代表一个系统的混乱程度，熵越大，系统越混乱。当一组内所有的变量值都一样的时候，熵值为零。因此，当我们分割一个集合时，其中一组与另一组的差别越大，同时每一组内的个体之间的差别越小，也就是说，对一个数据集中数据的分组，就是使得该数据集熵减小的过程。如果我们使用一种方法对数据集进行分组，如果熵值并没有下降，那么在很大程度上证明，这不是一个好的特征，或是一个好的分割值。

如果我们将一棵已经创建的决策树进行倒置，那么当我们组合两个相同位置的节点时，熵变化会随之增加，熵的增加称为信息增益。 所以在颠倒后，信息增益越高，分割也越好。

这样看似逻辑性不强，但却具有非常强的数学基础做支撑，这种数学基础被用于创建建模软件，来构造决策树。

当给定一组具有许多特征的样本时，决策树将识别最佳分割的特征以及用于分割的特征值。 然后，它将输出一系列参数，包括被用于分割的特征数量，具体特征值和信息增益等。

那么建模软件如何创建最好的决策树呢？ 它一个一个地试，并比较每个的信息增益，然后对比选择出最好的。

**介绍一下随机森林**

决策树提供了一个简单、清晰的概念模型来理解迭代分类过程。然而，在实践中，单个决策树对于解决涉及大量变量和中等大小规模数据的现实问题时，并不十分有效。因此，我们需要重型武器——由一群“专家”组成的决策森林。

这里的“专家”指的是什么呢？

我们认为的“专家”们，每人脑子里有一棵决策树模型，然后将这样的100个专家集合在一起，不那么严格地讲，我们就创建了一个随机森林。当我们期望决策结果好于单个决策树的时候，我们会想要一群“专家”来帮忙做决策。因此，我们需要一些方法来决定如何整理和排序“专家”们的“意见”。

回想一下奥林匹克体育比赛或潜水比赛，那里有一组评委对每一名参赛者打分，你可能会记得，分数中的最高分和最低分将被去掉，其余的求平均。 随机森林算法使用这种技术来去除一些“意见”，但可能会随机去掉一定比例的“意见”，然后重新运行“比赛”，重复进行此操作，比方说 100次，然后对结果求平均。

**为什么要有这么复杂的技术呢？**

一方面，数学上证明，他们是更为精确的。而且因为当我们拥有10倍或者100倍的特征数时，随机森林能够筛选出最重要的特征，并且将这些特征应用于建模中，而且不需要我们做任何额外的努力。那么这是如何做到的呢？这涉及计算成本的问题，所以我们的模型可能会运行几分钟，而不是几秒钟来处理成千上万个样本，目前大规模的计算已成为现实。然而，这一个成本还带来了其他的好处，

随机森林能更好地处理缺失值、噪声和异常值，并且可以处理数量变量和分类变量的混合型数据类型。

我们将探索从手机加速度计量仪和陀螺仪获取的丰富数据集，并基于对数据特征的特定组合，来了解用户的各种活动，如坐着，站立，步行等。我们的数据集有500多这样的特征，数据复杂、多样，所以这是一个将专业知识与随机森林力量相结合、进行探索和分析的较佳选择。

**练习**

学习下列术语，并研究他们如何将决策树与随机森林相关联的。

* 自助抽样
* 自助聚类
* Bagging算法
* Boosting算法
* 袋外错误

参考

[1] 决策树 <http://en.wikipedia.org/wiki/Decision_tree_learning>

# C2随机森林 - 数据预处理

简介

现在，我们应用随机森林对一个从熟悉的源对象获取的庞杂数据集进行分析。为什么要使用随机森林？马上你就会清楚。

我们将使用从手机加速度器和陀螺仪获取的数据集来创建预测模型。数据集的格式为R语言[1]，存在于Amazon S3，元数据存储在UCI数据库中[2]。数据集读取了手机方位和手机携带者运动的编码数据。

观察对象从事以下六种活动中的一种 - 坐着，站立，躺下，漫步，往前走，往后走。

**方法**

我们的目标是，通过一个给定的数据点，能够预测出他们正在从事的活动。我们设定了一个目标，即利用可以理解的变量来创建一个模型，而不是一个“黑匣子”模型。我们完全可以选择创建一个只有变量和系数的黑盒子模型。当给定数据点时，我们输入模型，其将弹出一个答案。这通常是可行的，但只是有太多“魔术”的成分，对于我们建立直觉判断或应用专业知识没有太多的帮助。

所以我们打算打开这个盒子，一旦我们形成一些直觉，我们就可以将专业知识和随机森林的大量功能结合起来。我们发现，从长远来看，这是一个更令人满意的方法，而且看起来非常强大。

我们将单独使用专业知识将自变量的数量减少到36个，然后使用随机森林来预测“活动”变量。从准确性的角度来看，这可能不是最好的模型，但是我们想要了解发生了什么，从这方面来讲，事实证明是更佳的。

我们使用准确度指标——正负估计值，敏感度和特异性来评估我们的模型。

**清理数据**

给定的数据集包含21名受访者的活动数据。

数据集有7,352行，561个数字列，外加2列整数表示的“主题”列和字符串表示的“活动”列。

由于共有563列，我们将省略创建正式数据字典的步骤，直接引用feature\_info.txt中的内容。

通过初步分分析发现，列名称文本含糊且存在许多问题：

* 列名重复-多次出现
* 列名包含（）
* 许多列名存在多余的“）”
* 列名包含“-”
* 列名包含“，”
* 许多列名包含“BodyBody”，我们假设这是一个打字错误。



将“活动”转换成一个分类变量，

将“受访者”设置为整数类型

创建一个可解释的模型，而不是将随机森林作为一个黑盒子来使用。 所以我们需要了解我们的变量，并提升我们对它们的直觉。

为了规划数据挖掘，UCI网站[2]数据集的文档非常有用，我们对其进行了详细的研究。 特别是文档feature\_info.txt，它对我们理解这些变量非常重要。 实际上，我们拒绝使用数据字典，而是使用了feature\_info.txt中对术语的解释。 因此，先回顾一下其中的细节非常关键。

**练习**

对每个数据集进行上述清理活动。即

识别和删除重复的多次出现的列名称

识别和修正列名中列出的（）。 你会如何解决这个问题？

识别和修改列名称中多余的）。 你会如何解决这个问题？

识别并修改列名中的“ - ”。 你会如何解决这个问题？

在列名中识别并修改多余的“，”。 你会如何解决这个问题？

识别并修改包含“BodyBody”的列名

**对学习者的提示**

该数据集的主要价值如下

它让我们领会到：

你可以仅使用强大的“黑盒”技术，然后获得有用的结果

你也可以利用专业知识来简化流程。此数据集能够突显你在专业知识上获得的优势。

它让我们走出舒适区，从语义相近的数据源中寻求相关信息，以进一步增强分析能力。

这说明了一个事实，就是你很难将数据和所有相关的信息整合在一起。

我们早就知道要对数据进行清理，但是我们更想扩大我们的知识面，做一点研究来增强我们的专业能力。

所以这个特定的数据集可能看起来有点技术性，但它很容易就与生物学，或者金融、力学相结合。一位数据科学家应该愿意把自己搞的有点手忙脚乱。最成功的科学应该是愿意跨学科的人。

这就是我们所领悟到的。

除了理解每个变量的含义，我们还想得到关于每个变量含义的技术背景。

所以我们使用Android开发人员指南[3]来自学每一个重要的物理参数。以这种方式，我们扩展专业知识，以便我们了解数据语言-我们想让它活着，对我们说话，并从中获知它的奥秘。我们对数据来源的背景了解的越多，那么我们对数据的探究就越好，越快，越深入。

在这种情况下，带有 X，Y，Z前缀/后缀的变量以及Android开发人员指南[3]给出了衡量变量的具体参考框架，包括加速度、加速度的矢量分量、重力角，或者更准确地说是由重力引起的向量加速度。我们在使用这些信息的同时，结合了我们对运动，速度，加速度的直觉。

**变量压缩**

所以我们对变量进行了挖掘，并做了一些简短的注释。

在进一步探究之前，你需要打开数据集目录中的HAR文件夹。有一个名为feature\_info.txt的文件。该文件描述了每个特征的物理含义，以及通过对原始数据进行求平均、抽样等操作后得出的数值。

让我们看一下：

a）所有变量名

b）物理量

首先让我们了解一下这些内容。

一旦我们投入时间开始做，就可以结合对人体的理解和常识，得到一些有用的指南。

* 在静态活动（坐，站立，躺下）中，运动信息不会很有用。
* 在动态活动中（3种类型的行走），运动信息将会很重要。
* 角度变量在区分“躺与站立”和“向前走与向后走”方面，将非常有用。
* 加速度和压力变量在区分各种运动时是重要的，同时过滤掉静止状态下的随机震颤。
* 大小和角度变量包含了与XYZ变量（=强相关）含有相同的信息
* 我们选择专注于后者，因为它们更便于解释。
* 这一点非常重要，因为它缩减了几百个变量。
* 我们忽略频率变量，因为我们无法做出简化和解释，并将其与物理活动相联系。
* 平均值和标准差非常重要，我们也将偏度和峰值包括在内。
* 将图1中变量作为预测因子是有效的，它们展示了我们的探究工作并验证了我们的想法。

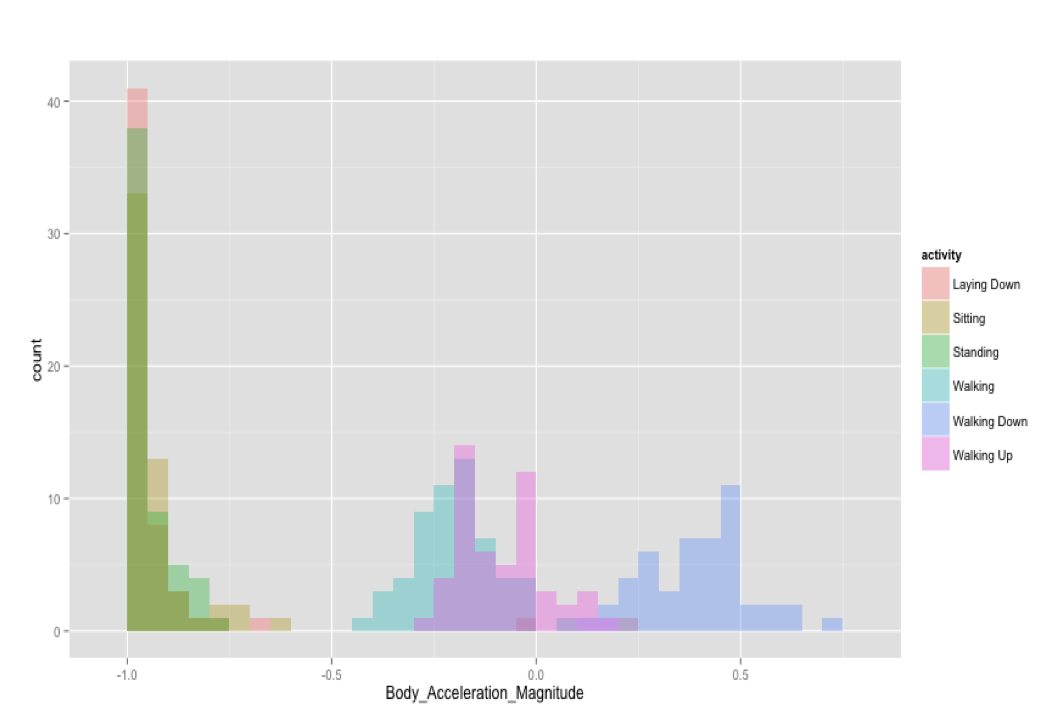


图1.使用身体加速度的直方图来评估作为静态和动态活动的预测因子的变量。 这是一个我们使用专业知识进行变量选择和数据探究的一个例子。

**消除干扰**

在缩减-X-Y -Z变量（笛卡尔坐标）时，我们删除了大量的干扰变量，因为它们与极坐标的大小和角度强相关。 可能仍然会有一些干扰，但剩余的影响不会很大。

从常识来看，由于最小值，最大值等与平均值/标准差有相关性，所以我们也删除所有这些干扰。最终变量的数量减少到37，如下：

**阅读提示：**我们做了一些繁琐的名称映射，以保持语义完整，因为我们想要一个“白盒子”模型。我们可以用几行代码和注释，将剩余的变量映射到v1，v2 ..... v37。但是我们将失去利用专业知识进行解释的，并从中获取价值的机会。

**名称转换**

为了方便研究数据，我们对变量进行了重命名和简化，以方便阅读，如下所示：

我们去掉所有的“body”和“Mag”，然后将“mean”替换成“Mean”，将“std”替换成“SD”。

例如

tAccBodyMag - - > tAccMean

fAccBodyMag-std - > fAccSD

等等

结果

缩减后，已转换名称的选定变量（具有有意义的分组）：

tAccMean, tAccSD tJerkMean, tJerkSD

tGyroMean, tGyroSD tGyroJerkMean, tGyroJerkSD

fAccMean, fAccSD, fJerkMean, fJerkSD,

fGyroMean, fGyroSD, fGyroJerkMean, fGyroJerkSD,

fGyroMeanFreq, fGyroJerkMeanFreq fAccMeanFreq, fJerkMeanFreq

fAccSkewness, fAccKurtosis, fJerkSkewness, fJerkKurtosis

fGyroSkewness, fGyroKurtosis fGyroJerkSkewness, fGyroJerkKurtosis

angleAccGravity, angleJerkGravity angleGyroGravity, angleGyroJerkGravity

angleXGravity, angleYGravity, angleZGravity

subject, activity

**结论**

现在，经过处理后，我们得到了完美的结果。

获得的主要变量为加速度的大小、抖动统计量和角度变量。 这让我们感受到，专注于专业知识，进行一些额外的阅读、研究，并结合一些基本的身体直觉，收获将非常大。

这是一组语义紧凑，可解释和相对容易理解的变量。

我们可以对变量进行新一轮的研究，我们可能认为，37个变量太多了，无法一次性记住 。为了更好地解决这个问题，我们使用重型武器，借助建模软件和随机森林算法来处理。

**参考**

[1] <https://spark-public.s3.amazonaws.com/dataanalysis/samsungData.rda>  
[2] Human Activity Recognition Using Smartphones <<http://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones>  
[3] Android Developer Reference <http://developer.android.com/reference/android/hardware/Sensor.html>  
[4] Random Forests <http://en.wikipedia.org/wiki/Random_forest>  
[5] Code for computation of error measures <https://gist.github.com/nborwankar/5131870>



**C3随机森林-分析**

**简介**

分析阶段的目标是用探究阶段处理后的变量数据集来构建一个随机森林，以预测人类的活动。

这些变量如下：

* tAccMean, tAccSD tJerkMean, tJerkSD
* tGyroMean, tGyroSD tGyroJerkMean, tGyroJerkSD
* fAccMean, fAccSD, fJerkMean, fJerkSD,
* fGyroMean, fGyroSD, fGyroJerkMean, fGyroJerkSD,
* fGyroMeanFreq, fGyroJerkMeanFreq fAccMeanFreq, fJerkMeanFreq
* fAccSkewness, fAccKurtosis, fJerkSkewness, fJerkKurtosis
* fGyroSkewness, fGyroKurtosis fGyroJerkSkewness, fGyroJerkKurtosis
* angleAccGravity, angleJerkGravity angleGyroGravity, angleGyroJerkGravity
* angleXGravity, angleYGravity, angleZGravity
* subject, activity

对于这些，

* 除了最后两列，其他所有都是数值型数据。
* “受访者”是一个整数，用于标识单个的人，其中从1到27的21个数据集有缺失项。
* “活动”是一个分类变量 - 早期确定的六个活动，包括 “坐”，“站立”，“躺”，“漫步”，“向前走”，“向后走”。
* 为什么我们使用随机森林？ 我们在模型中使用随机森林[4]，因为这种方法具有相对较高的精度，其次我们的数据太过复杂。

这是我们用随机森林的两个主要原因，特别是在压缩后，特征数量依然很大。

**方法**

**实验设计实例**

通常在分析这些数据集时，我们使用这些数据来创建一个模型。我们如何知道该模型同样适用于其他数据呢？真实的答案是“我们不知道”。而且我们无法确知我们可以创建一个同样适用于新数据的模型。

但是我们如何做才能降低模型“过度拟合”的概率呢？“过度拟合”是一个技术术语，它表示，模型在给定的数据（适合它）有完美的表现，但对新的数据集，却表现很差。有一种建模方法来避免这种陷阱。如下所述。

我们从数据集中留出部分数据用于测试，与建模用的数据集相区别，这个“保留”的数据集称为测试集。

然后我们取剩余的数据，对它进一步划分，较大的数据集称为训练集和较小的集称为验证集。然后，我们使用训练集构建我们的模型，并查看它在验证集上的表现（这里不包括“保留”数据）。

我们可以随便使用训练集和验证集来调整我们的模型，在模型建好之前，不能使用“保留”的测试集。建模完成后，我们将模型运行“保留”的测试数据，当测试数据达到可接受的错误率时，我们就完成了模型的创建。

然而，如果模型在运行测试集时，显示较高的错误率，那么这个模型就有问题。我们不能继续调整模型以获得更好的测试结果，因为过度拟合会再次出现。那么我们该怎么办？我们将所有的数据混合，然后“保留”新的测试集，然后重复运行。在某些情况下，我们的测试集将是第三方的数据，我们在交付模型前，是无法利用测试集进行测试验证的。我们必须一次性成功。

**我们的实验设计**

我们将数据中的最后4个被访者的数据作为测试集，其余用于建模。我们为什么要做这个？如果我们查看支持文档，也是建议使用最后4个作为测试集。不妨，我们开始就按照建议来做。我们保留了与这4个相关的所有的行，在建模时将它们排除在外。

在剩下的17个“受访者”中，我们使用前12个“受访者”作为训练集，剩下5个作为验证集。为什么是这个比例？通常，30％的数据用作验证集，70％用于实际训练。验证集用作我们的“内部”测试，不用于建模，并且每一步都会用到。实际测试集和验证集之间的区别在于，只要我们在每次运行软件后，混合数据，然后重新提取验证集，就可以继续调整优化我们的模型。

还有一种方法可以在此基础上，更进一步，能够进行n次验证。将训练集合分为n（通常为10）等份，然后将每个部分依次用作验证集，其余部分用于训练，并进行n次此类建模操作。然后将这些模型求平均，以创建最好的模型。

我们在这里不做n次验证。

因为我们的模型中包含“受访者”，因此，我们根据“受访者”变量来划分数据，并希望将所有测试数据独立出来。这是什么意思？测试数据实际上应该对我们而言，是完全未知的数据 - 即它需要独立于训练数据。所以假设我们没有分离出4个测试人员的数据，但是我们只是混合了所有的行，并随机选取20％作为测试数据集。

请注意，我们有大约7,000行，所以我们每个人都有几百行。如果我们将它们混合并随机选择，那么我们的测试集将包含21个人的信息，当然训练集中同样也会包含21个人的信息。测试集不会独立于训练集，因为两者的数据存在某种程度的重合。因此，被提取的集合不会实现真正有用的验证效果 - 统计学上，我训练集和验证集已经具有相似的信息，即测试集已经泄漏到训练集中。

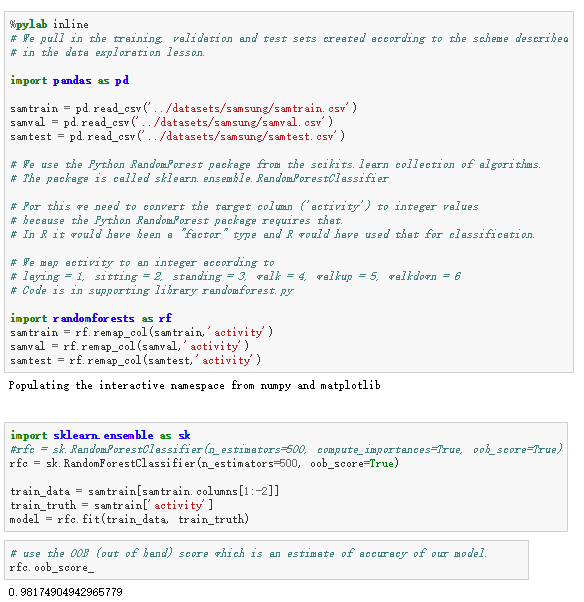
这与我们某次家庭作业练习的情况类似，后来第二天在课堂上解决了。然后我们收到一组问题，与作业练习非常相似，只是一些数字改变了。它不会真正地验证我们对核心问题的理解，只是让我们更加了解作业（过度拟合）。

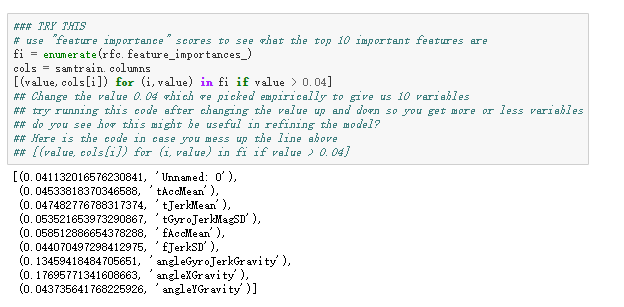
所以当我们将部分人的行数据独立出来后，我们知道训练集并没有泄露这些人的信息。对测试数据做这样的努力是非常值得的，所以我们可以很有信心地讲，我们的模型不会过度拟合样本数据。

**结果**

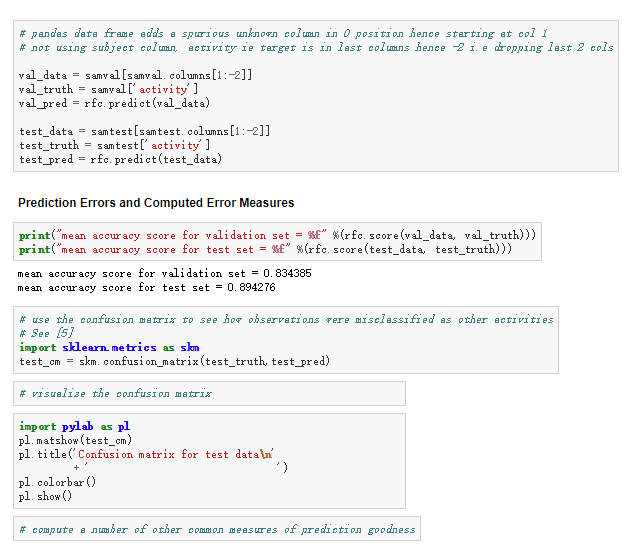
**训练**

我们在随机森林的建模软件上运行我们的数据，如前所述，并推导出一个模型以及一些描述我们的模型有多好的参数。





**应用模型中预测函数运行验证集和测试集，然后对误差进行分析，得到以下结果：**



**我们现在应用一些常用方法来预测“好”。**

**有关这些措施的更多细节见[6]，[7]，[8]，[9]**



**结论**

我们可以从以上结果中得出以下具体结论。

在这种情境下，随机森林的表现令人非常满意的，较低错误率和以及较高预测能力。

使用专业知识可以获得令人惊讶的高预测度量值，以及验证集和测试集上的低错误率。

结果支持，即预测准确率达到90%，OOB误差率约2％。

我们只做了一次，并没有来回调整模型。请注意，我们坚持这里的规则，直到完成建模后才拿到测试集。

将加速度的大小、角度以及抖动的时间和幅度作为变量，构建出了预测能力非常好的模型。当然，暴力模型可以提供更好的预测能力，但它是一个黑盒应用软件。如果由于某种原因造成的模式错乱或失败，我们根本就不了解为什么。相反，我们使用专业知识来观察这个过程，创建了模型是可以理解和解释的。

运行测试时，模型在测试集上的表现优于验证集，数据无论来源于“汇总”行，还是单独某个人的。

让我们看一下今后如何改进模式。注意我们的模型在某些情况下，可能是存在缺陷，或是不完整，待改进等，所以我们不会对模型过分自信，并且做能力之外的事情。

**点评**

* 利用专业知识，模型在变量选择过程中消除了一些与力的大小相关的特征，如最大值、最小值以及一些与旋转相关变量。 这些可能很重要，但未经测试。 我们可能会在下次进行观察。
* 应该详细研究变量重要性 - 即我们真的应该使用被认定更为重要的少量特征来来创建模型，看看有什么区别。 在计算上这将更有效率。 我们甚至可以使用简单的方法，如逻辑回归，对简化的数据集进行分类。

**练习**

不是使用专业知识来减少变量，而是直接在全部列上使用“随机森林”。 然后使用变量重要性并对变量进行排序。

将得到的模型与您使用专业知识获得的模型进行比较。

您可以通过简单地重命名变量x1，x2 ... xn，y来缩短数据清理过程，其中y是因变量的“活动”。

现在看看你得到的新的随机森林模型。 在预测中可能比我们以上方法更准确。 这是一个黑匣子模型，其中的变量没有什么意义。

它给了你什么见解？

你喜欢哪个型号？

为什么？

这是绝对的偏好还是可能会改变？

什么可能导致它改变？

**参考**

[1] Original dataset as R data <https://spark-public.s3.amazonaws.com/dataanalysis/samsungData.rda>  
[2] Human Activity Recognition Using Smartphones <http://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones>  
[3] Android Developer Reference <http://developer.android.com/reference/android/hardware/Sensor.html>  
[4] Random Forests <http://en.wikipedia.org/wiki/Random_forest>  
[5] Confusion matrix <http://en.wikipedia.org/wiki/Confusion_matrix> [6] Mean Accuracy <http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=1054102&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D1054102>

[7] Precision <http://en.wikipedia.org/wiki/Precision_and_recall> [8] Recall <http://en.wikipedia.org/wiki/Precision_and_recall> [9] F Measure <http://en.wikipedia.org/wiki/Precision_and_recall>



1. 利用随机森林进行分类和回归 [↑](#footnote-ref-1)