建立学生干预系统

主要内容：使用pandas，numpy和pyplot探索数据，通过将非数值数据转换为数值数据对数据进行预处理，使用适当的监督学习算法（如朴素贝叶斯，逻辑回归和支持向量机）进行预测，并计算和比较F1分数。

文章来源：翻译自《Building a Student Intervention System》

http://www.ritchieng.com/machine-learning-project-student-intervention

作者简介：本文作者Ritchie Ng是来自新加坡国立大学（NUS）的深度学习研究人员、全球优秀学者，专门从事深度学习、计算机视觉和自然语言处理方面的研究，其研究项目涉及TensorFlow，PyTorch，TensorLayer，Keras，OpenCV，scikit-learn，CUDA，cuDNN，Python以及C / C++等相关内容。

作者网站：http://www.ritchieng.com

# 1 项目内容：建立学生干预系统

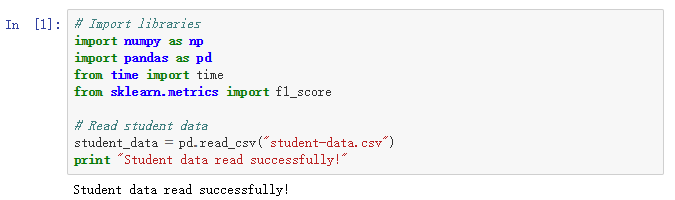
在这个项目中，我们的目标是根据已提供的学生个人数据以及往届学生的毕业情况，找出学生各项指标和学生能否毕业之间的关系。划分出训练集和测试集，对建立的模型进行训练与测试，根据各项指标将学生分为“可能毕业”与“可能不会毕业”两类，从而确定是否需要对学生进行提前干预。

本项目使用的数据集在student-data.csv中，数据包括学生的基本信息（性别、年龄、健康状况等）、学生的家庭信息（家庭大小、父母受教育程度、父母工作情况等）、学生的学习情况（学校、每周学习时间、旷课次数、挂科次数等）以及学生的课余生活状况（是否有课外活动、放学后的课余时间、工作日及周末的酒精消费情况等）共计31个指标。其中，最后一列‘passed’（学生是否毕业）将是我们的目标标签，其他所有列是每个学生的特征指标。

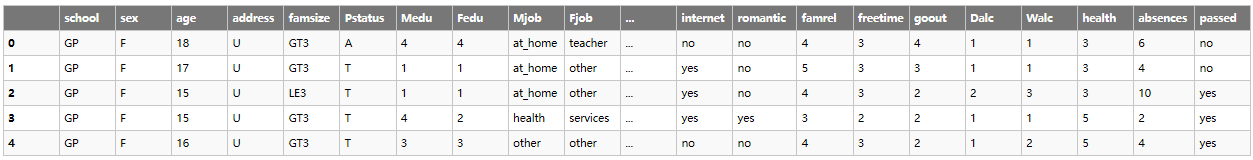
为了了解我们的项目内容，确定数据集的基本情况，获知往届学生的毕业率，我们将首先计算出我们所关注的内容：学生总人数、每个学生的特征总数、毕业的学生人数、未毕业的学生人数以及毕业率等，以此来了解学生数据的基本状况，以便我们在后续的数据准备中确定特征和目标列，并对特征列进行预处理，以实现训练集和测试集的拆分，为建立模型做准备。

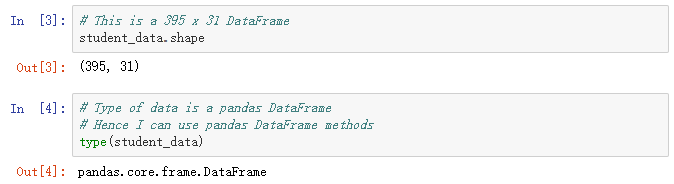
## 1.1 数据探索

运行下面的代码，加载必要的Python库和学生数据。请注意， 最后一列‘passed’（学生是否毕业）将是我们的目标标签，其他所有列是每个学生的特征指标。





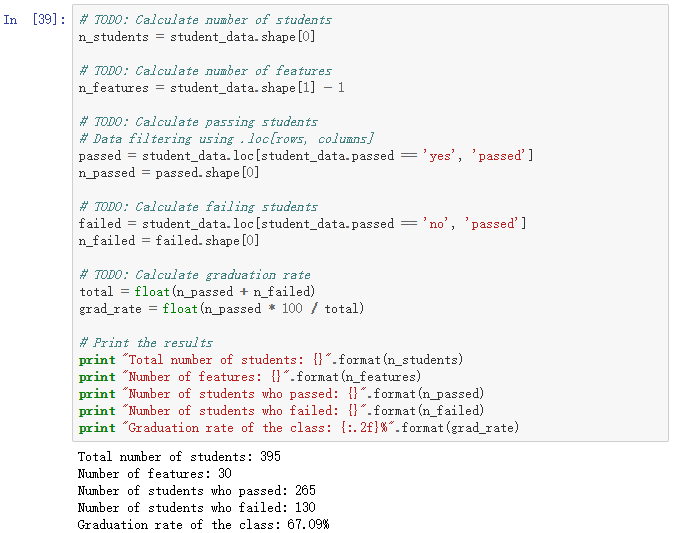




从上述代码的运行结果可以看出，该数据集共包含395名学生的31个指标，由于数据集类型是pandas数据类，我们可以直接使用pandas对该数据集进行处理。

为了确定我们有多少学生的信息，并了解这些学生的毕业率。在后面代码中，需要计算以下内容：

* 学生总人数n\_students。
* 每个学生的特征总数n\_features。
* 毕业的学生人数n\_passed。
* 未毕业的学生人数n\_failed。
* 毕业率grad\_rate，使用百分比（%）。



从输出结果可以看到，总学生数为395，特征指标为30（‘passed’被我们作为了目标标签），成功毕业的人数为265人，未能毕业的人数为130人，毕业率为67.09%。

### 问题1 - 分类与回归

在这个项目中，我们的目标是确定可能不能毕业而需要及早干预的学生。这是哪种类型的监督学习，分类还是回归？为什么？

### ****答案：****

* 这应该是一个分类问题。
* 这是因为它有两个离散的结果，这是典型的分类问题：
  1. 需要早期干预的学生。
  2. 不需要及早干预的学生。
* 我们可以用二进制结果进行相应的分类，如：
  1. 是，1，需要早期干预的学生。
  2. 否，0，不需要早期干预的学生。
* 由于我们并不想预测得到一个连续的结果，因此这不是一个回归问题。

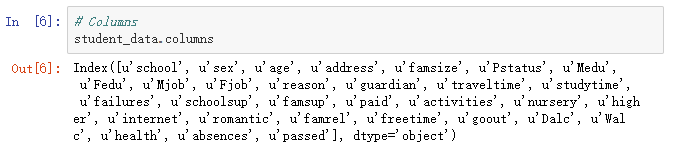
# 2 数据准备

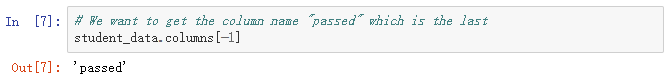
在这一部分，我们要准备用于建模、训练和测试的数据。

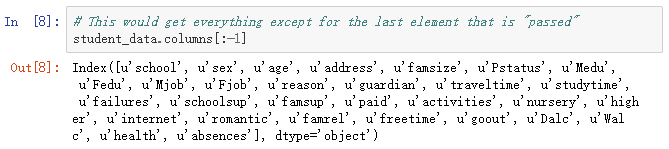
## 2.1 确定特征和目标列

通常情况下，我们获得的数据包含非数字类型的特征。这是一个比较麻烦的问题，因为大多数机器学习算法更希望使用数字类型的数据进行计算。

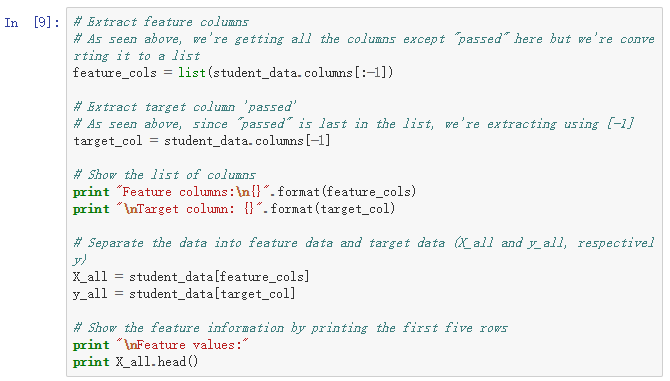
运行下面的代码将学生数据分成特征和目标列，查看是否有非数字类型的特征。

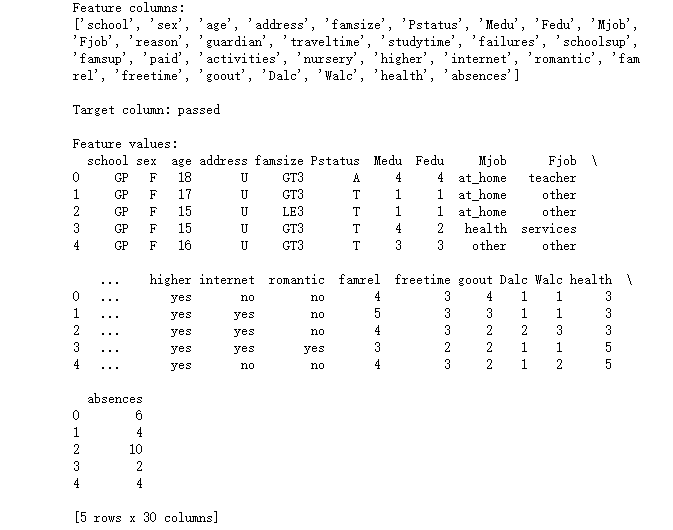






我们在此获得了除目标标签“passed”以外的所有列标签。接下来进行特征列的提取，首先将数据转换为一个表，将数据分为特征数据和目标数据。





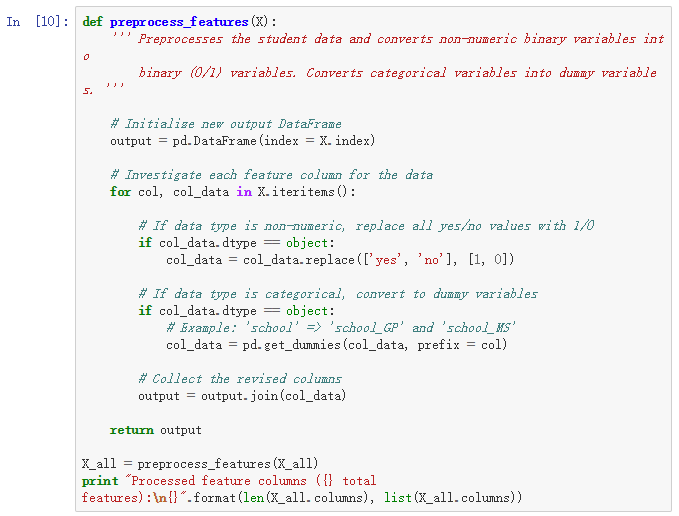
从打印结果可以看出，数据已经转换成表的形式，“passed”为我们的目标列，其余指标列为特征列。

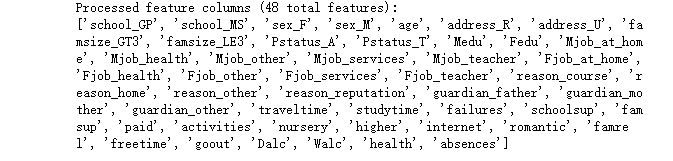
## 2.2 特征列预处理

可以看到，有几个非数字列需要转换。其中许多列只是yes/ no，例如internet。这些可以转换为1/ 0（二进制）值。

对于其他列，如Mjob和Fjob，具有两个以上的值，称为分类变量。处理此类列的建议方法是，创建与所有可能值的数量相同的列（例如Fjob\_teacher，Fjob\_other，Fjob\_services等），并将1分配给其中一列，0分配给其他所有列。

有时这些生成的列称为虚拟变量，我们将使用pandas.get\_dummies()函数来进行这一转换。运行下面的代码来执行这一部分所讨论的预处理程序。





通过以上代码，我们对数据进行了简单的处理，并成功将非数字二进制变量转换为了二进制变量，分类变量转换为虚拟变量。

## 2.3 拆分训练数据和测试数据

到目前为止，我们已将所有的分类特征转换为数值类型。下一步，我们将数据（特征和其相应的标签）分为训练集和测试集。在下面的代码中，需要实现以下内容：

* 随机重组，将数据（X\_all，y\_all）拆分为训练子集和测试子集。
  + 使用300个训练点（约75％）和95个测试点（约25％）。
  + 为所用的函数设置random\_state（如果提供了的话）。
  + 将结果储存在X\_train，X\_test，y\_train，和y\_test中。

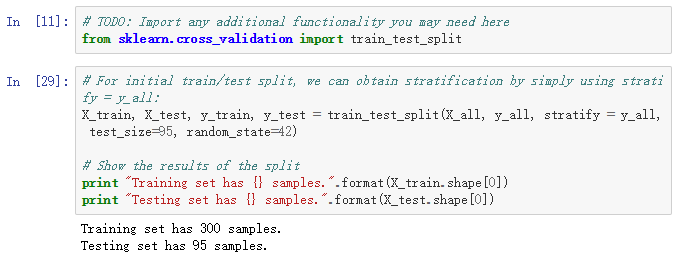
**特别提示：**数据评估对训练集/测试集的影响

在处理新的数据集时，较好的做法是评估其具体特征，并使用根据这些特征制定的交叉验证方法，在本例中有两个要点：

* 我们的数据集较小。
* 我们的数据集略有些不平衡。（已经通过的学生比正在通过的学生更多）

**我们能做什么？**

* 我们可以利用K-折交叉验证来探索小数据集
* 如果我们需要处理严重不平衡的数据集，即使在这种情况下可能没有必要，我们也还是可以使用分层K-折和分层随机分流交叉验证法来解决我们的数据集不平衡性，因为分层保留了每个类别的样本百分比
  + http://scikit-learn.org/stable/modules/generated/sklearn.cross\_validation.StratifiedShuffleSplit.html
  + http://scikit-learn.org/stable/modules/generated/sklearn.cross\_validation.StratifiedKFold.html





对于初始的训练集测试集分离，我们可以使用分层的方法得到分层数据。结果显示训练集有300个样本，测试集有95个样本，然后对分层结果进行双重检查。

# 3 模型训练与评估

在这一部分，我们将选择适合此问题并在scikit-learn中可用的3个监督学习模型。首先，根据对数据的了解和每个模型的优缺点讨论选择这三个模型的原因。然后，拟合不同大小的训练数据（100个数据点，200个数据点和300个数据点），并测量F 1分数。需要生成三个表（每个模型一个），用来显示训练集的大小、训练时间、预测时间、训练集的F 1分数，和测试集的F 1分数。

scikit-learn中有以下监督学习方法可供选择：

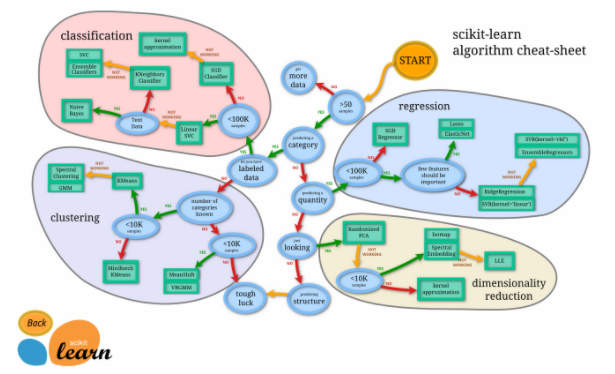
* 高斯朴素贝叶斯（GaussianNB）
* 决策树
* 组合方法（Bagging，AdaBoost，Random Forest，Gradient Boosting）
* K-近邻（KNeighbors）
* 随机梯度下降（SGDC）
* 支持向量机（SMV）
* 逻辑回归

### 问题2 – 模型应用

列出适合此问题的三个监督学习模型。对于所选择的每个模型

* 描述一下在可应用该模型的行业中的真实应用。（给出参考文献）
* 模型的优点是什么？何时表现良好？
* 模型的缺点是什么？何时表现不好？
* 根据你对数据的了解，是什么使该模型成为这个问题的良好备选项？

如何选择算法？



### 答案：

* 我们将选择3个监督学习模型。
  1. 朴素贝叶斯
  2. 逻辑回归
  3. 支持向量机
* 但在我们继续介绍3种监督学习模式之前，我们将讨论数据本身，因为它对于确定该模型是否成为当前问题的良好备选项是一个重要的因素。

#### 数据概述

##### 1. 分类偏斜：

* 可以看到，与未能毕业的学生相比，成功毕业的学生人数几乎是其两倍。
  + 成功毕业学生人数：265（多数类）
  + 未能毕业学生人数：130人（少数类）
* 这在我们分割数据时会产生问题。
  + 训练集可以用多数类聚集，测试集可以用多数类聚集。这将影响计算的精度。
  + 因此，应该强调如何拆分数据以及要选择的指标。
    - 数据分割：分层K折
    - 指标选择：准确率和召回率

##### 2. 数据缺失：

* 数据集中没有示例。
  + 395名学生
* 这将对需要更多数据的一些算法产生影响。
  + 一般来说，我们需要更多的数据，除非我们面临高偏差问题。
  + 在这种情况下，我们应该保持简单的算法。

##### 3. 特征过多：

* 对于这样一个有395名学生的小数据库，我们的特征数量惊人。
  + 48个特征
* 维度诅咒
  + 由于维度诅咒，对于添加的每个额外特征，我们需要增加示例的数量。

#### 模型说明

##### 1. 朴素贝叶斯

* 行业应用（Barbosa等，2014）
  + 鸡蛋分类
  + 这是朴素贝叶斯作为学习算法繁荣一个有趣的应用，将鸡蛋分为2组：
    1. 自由摆放的鸡蛋
       - 这些鸡蛋是能够自由滚动的鸡蛋
    2. 层架式摆放的鸡蛋
       - 这些鸡蛋是保存在一个小笼子里的鸡蛋。有些人可能称之为“unethical eggs”
  + 研究显示，在2组鸡蛋之间进行分类时，朴素贝叶斯提供了90％的高精度。
* 优点
  1. 由于比逻辑回归之类的判别模型收敛更快，因此需要更少的数据
* 缺点
  1. 要求观察值之间相互独立
     + 但实际上，即使违反了独立性假设，分类器也表现良好
  2. 表示简单，没有超参数调整的机会
* 问题的适合性
  1. 该算法对此问题表现良好，因为数据具有以下属性：
     + 观察值数量少
     + 朴素的贝叶斯在小数据集上表现良好

##### 2. 逻辑回归

* 行业应用（Penderson等，2014）
  + 蛋白质序列分类
    - 识别并自动将蛋白质序列分为11个预定义类别之一
    - 逻辑回归的使用在未来生物信息学的应用中有巨大潜力
* 优点
  1. 有许多方法用来规范模型以容忍错误并避免过度拟合
  2. 与朴素贝叶斯不同，我们不用担心有相关性的特征
  3. 与支持向量机不同，我们可以使用在线梯度下降法轻松获取新数据
* 缺点
  1. 需要观察值是相互独立的
  2. 其目的是基于自变量进行预测，如果没有正确识别，逻辑回归几乎没有预测价值
* 问题的适合性
  + 许多特征可能是相关的
    - 与朴素贝叶斯不同，逻辑回归可以处理这个问题
    - 由于数据集具有许多特征，正则化以防止过度拟合

##### 3. 支持向量机（SVM）

* 行业应用（Di Pillo等，2016）：
  + 促销活动的销售预测
    - 最初使用的统计方法是如ARIMA和指数平滑这样的平滑方法
    - 但如果出现销售数据的高无规律性，可能会失败
    - 因此SVM提供了一个很好的选择
* 优点
  1. SVM能正则化参数以容忍一些错误并避免过度拟合
  2. 内核技巧：用户可以通过设计内核来嵌入有关该问题的专业知识
  3. 如果选择参数C和γ，则样本推广效果好
     + 换句话说，即使训练样本有一些偏差，SVM也会更稳健
* 缺点
  1. 不好解释：支持向量机是黑箱
  2. 高计算成本：SVM的训练时间呈指数级增长
  3. 用户需要具有一定的专业知识才能使用内核函数
* 问题的适合性
  + 许多特征可能是相关的
    - 由于数据集具有许多特征，正则化以防止过度拟合

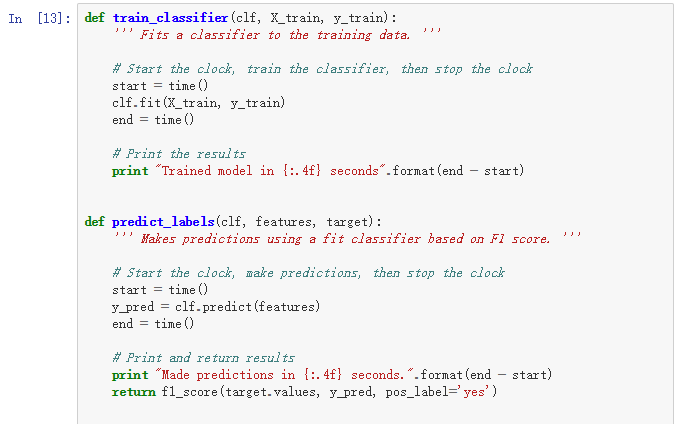
##### 参考文献

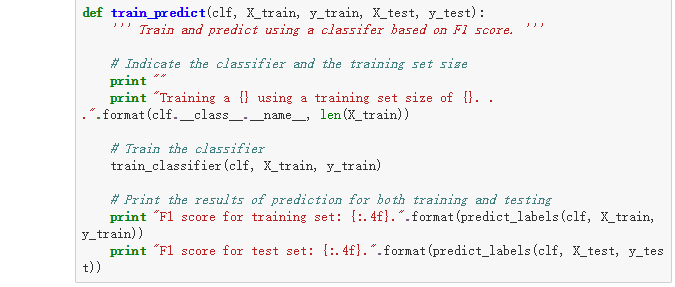
1. Barbosa, R. M., Nacano, L. R., Freitas, R., Batista, B. L. and Barbosa, F. (2014), The Use of Decision Trees and Naïve Bayes Algorithms and Trace Element Patterns for Controlling the Authenticity of Free-Range-Pastured Hens’ Eggs. Journal of Food Science, 79: C1672–C1677. http://dx.doi.org/10.1111/1750-3841.12577.
2. Pedersen, B. P., Ifrim, G., Liboriussen, P., Axelsen, K. B., Palmgren, M. G., Nissen, P., . . . Pedersen, C. N. S. (2014). Large scale identification and categorization of protein sequences using structured logistic regression. PloS One, 9(1), 1. http://dx.doi.org/10.1371/journal.pone.0085139
3. Di Pillo, G., Latorre, V., Lucidi, S. et al. 4OR-Q J Oper Res (2016) 14: 309. http://dx.doi.org/10.1007/s10288-016-0316-0

#### 设置

运行下面的代码以初始化三个辅助函数，可以使用它们来训练和测试上面选择的三个监督学习模型。函数如下：

* train\_classifier - 输入分类器和训练数据，并将分类器与数据进行匹配。
* predict\_labels- 将合适的分类器、特征和目标标签作为输入，并使用F 1分数进行预测。
* train\_predict- 输入分类器、训练数据和测试数据，并展示train\_clasifier和predict\_labels结果。
  + 此函数将分别汇报训练数据和测试数据的F 1分数。



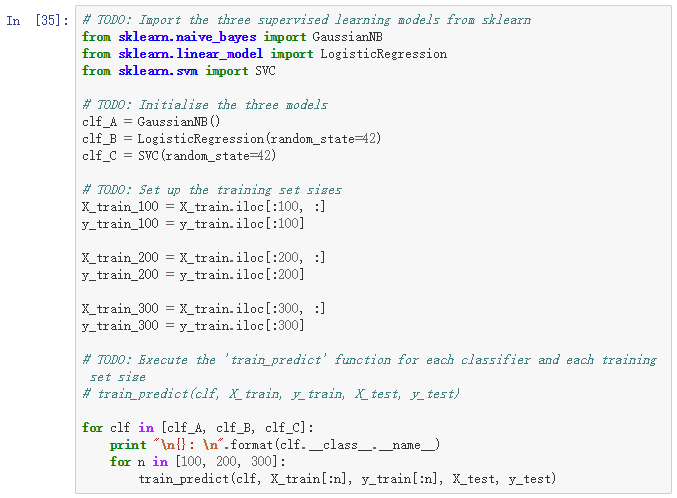


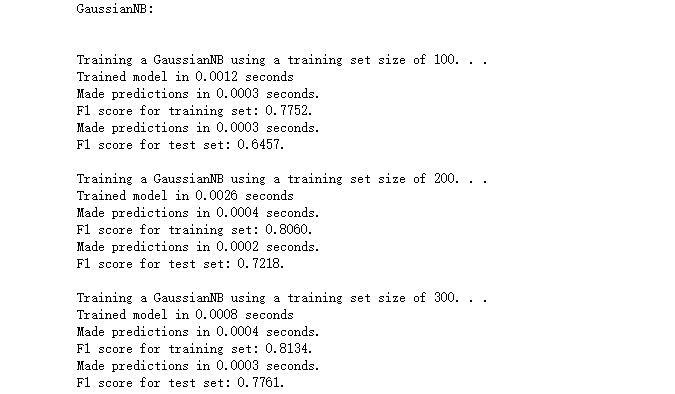
其中，train\_classifier函数用分类器拟合训练数据，prediction\_labels函数使用基于F1分数的拟合分类器进行预测，train\_predict函数使用基于F1分类器进行分类和预测。

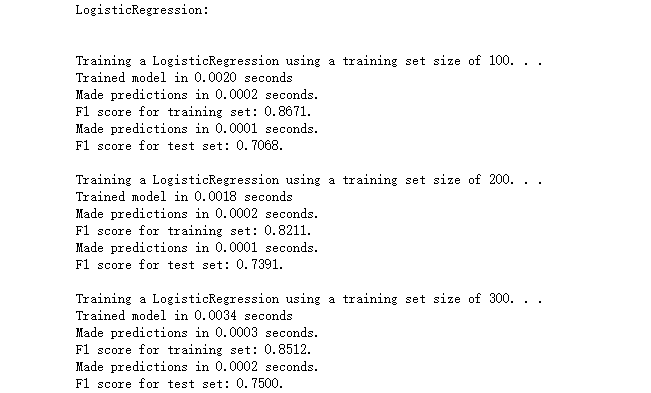
## 3.1 模型效果指标

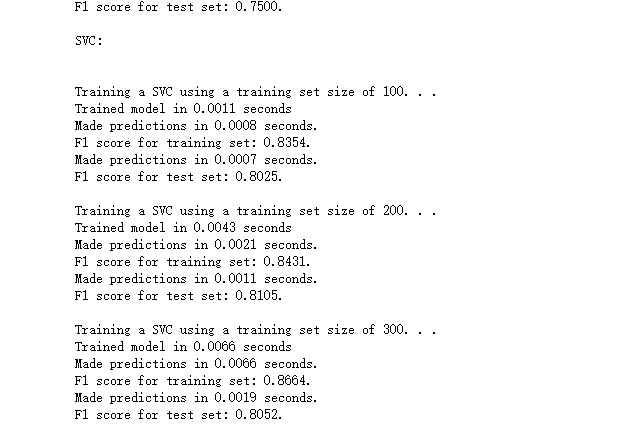
使用上面的预定义函数，导入你选择的三个监督学习模型，并为每个学习模型运行train\_predict函数。记住，你需要对每个分类器进行训练和预测，以获得三种不同大小的训练集：100、200和300。因此，你应该为每个模型使用不同的大小的训练集得出以下9个不同输出。在下面的代码中，需要实现以下内容：

* 导入你在上部分中选择的三个监督学习模型。
* 初始化三种模型，并把它们存储在clf\_A、clf\_B和clf\_C中。
  + 如果提供random\_state，为每个模型使用此函数。
  + **注意：**为每个模型使用默认设置 - 在稍后的部分中你会调整一个特定的模型。
* 创建用于训练每个模型的不同训练集大小。
  + 不要重组并重新分配数据！新的训练点应来自X\_train和y\_train。
* 将每个模型与每个训练集大小相匹配，并对测试集进行预测（总共9个）。  
  **注意：**以下代码之后提供了可以用于存储结果的三个表。





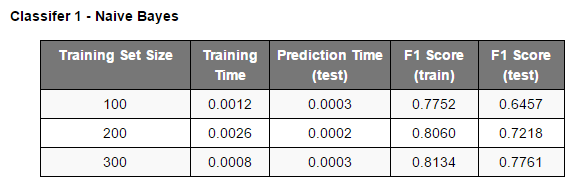


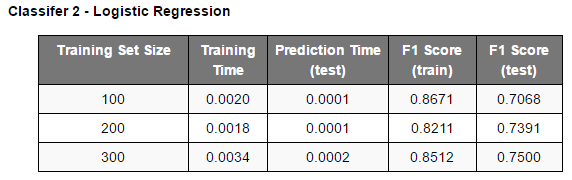


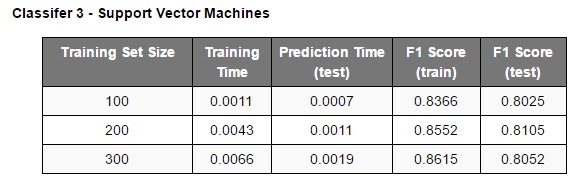
从输出结果可以看出，不同算法、不同数据集大小所得的结果不同，为了进行对比，我们将结果存放到表格中。

## 3.2 表格结果

在Markdown中设计表格，编辑下面的单元格，可以在表中记录上述结果。







从三张表中，可以很清晰的看出三种不同算法对应三种不同大小数据集所用的训练及预测的时间和F1分数，经过对比和分析，我们可以选则一个最合适的模型。

# 4 选择最佳模型

在最后一部分中，我们将从三个监督学习模型中选择用于学生数据的最佳模型。然后，对整个训练集（X\_train和y\_train）的模型进行网格搜索优化，通过调整至少一个参数来改善未经训练模型的F 1分数。

### 问题3 – 选择最佳模型

根据之前的实验，用一到两段说明你所选择的最佳模型。基于已有的数据、有限的资源、代价和表现，哪种模式是最合适的？

### 答案：

SVM的预测性能比朴素贝斯略好，但需要注意的是，数据增加时，SVM的计算时间将会比朴素贝叶斯所需时间增加的快得多，当我们学生数量更多时，代价将会呈指数级增长。另一方面，朴素贝叶斯的计算时间随着数据的增加而线性增长，代价不会快速上升。因此，考虑到其在一个小数据集和潜在的大型和不断增长的数据集上的表现，朴素贝叶斯是SVM一个很好的替代。

再比较朴素贝叶斯和逻辑回归。虽然结果显示，逻辑回归在预测效果方面比朴素贝叶斯稍差，但与朴素贝叶斯相比，逻辑回归模型的轻微调整很容易产生更好的预测效果，而朴素贝叶则不能对模型进行调整。因此，我们应该选择回归。

3个算法的大O符号

1. 朴素贝叶斯：
   * O(n)
2. 逻辑回归：
   * O(C^n)
3. 支持向量机：
   * O(n^3) 关于S型核
   * O(n^2) 关于空间复杂性

### 问题4 – 模型的外行解释

用1-2段内容，向外行人士解释，最终选择的模型是如何起作用的。确保你描述的是模型的主要内容，例如模型如何训练以及模型如何进行预测。避免使用数学或技术术语，例如描述方程或讨论算法实现。

### 答案：

 首先，模型学习学生的表现指标对学生是否毕业的影响。由于我们已经有已毕业和没有毕业的学生的现有数据，所以该模型可以做到这一点。根据学生的表现指标，该模型将输出每个表现指标的权重。

考虑到这一点，第二步是预测新生是否毕业。这一次，我们没有关于现有学生的信息，他们是否毕业或还在读书。但是，我们有一个从以前批次毕业学生中学习到的模型。新增学生的表现指标及其权重将被输入模型，模型将产生概率，根据“可能毕业”或“不太可能毕业”将学生进行分类。然后，我们可以对不太可能毕业的学生采取预防措施。

此外，如果我们想要结果更加安全可靠，以确保我们发现尽可能多的学生，我们可以提升“可能毕业”的严格程度，将一部分学生判定为“可能不毕业”，即使他们可能“能毕业”，来确定他们毕业的可能性并找到更多学生。这是因为将“可能毕业”的学生标记为“不太可能毕业”没有任何害处，但是如果将“不太可能毕业”的学生判定为“有可能毕业”而不进行干预的学生，可能会导致严重的后果。

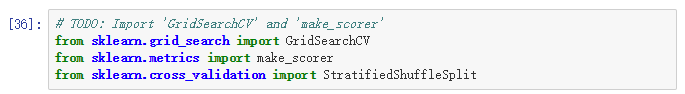
## 4.1 模型调整（逻辑回归）

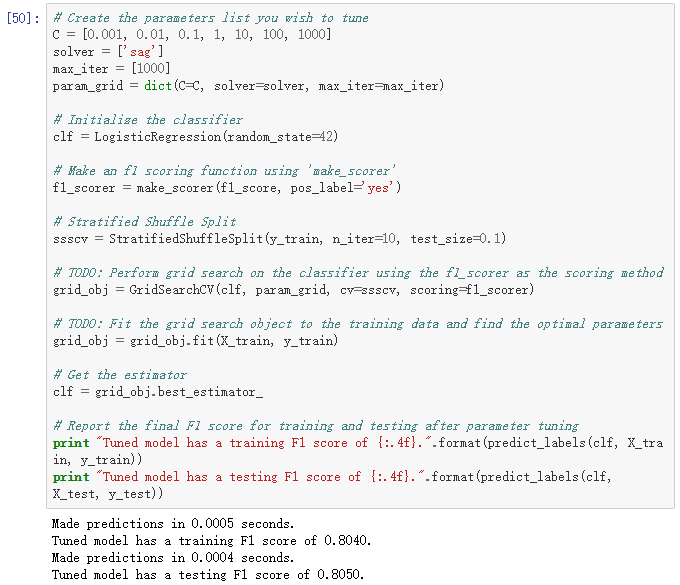
微调所选模型。使用网格搜索（GridSearchCV），其中至少有一个重要参数并调用至少3个不同的值，对整个训练集使用这一方法。在下面的代码中，需要实现以下内容：

* 导入sklearn.grid\_search.gridSearchCV和sklearn.metrics.make\_scorer。
* 为所选模型创建一个你希望调整的参数的字典。
  + 示例：parameters = {'parameter' : [list of values]}。
* 初始化已选择的分类器并将其存储在clf中。
* 用make\_scorer创建F 1评分函数并存储在f1\_scorer中。
  + 将pos\_label参数设置为正确值！
* 使用f1\_scorer评分方法在分类器clf上执行网格搜索，并将其存储在grid\_obj中。
* 将网格搜索对象与训练数据（X\_train，y\_train）进行拟合，并将其存储在grid\_obj中。

**特别提示：**

* 可以使用分层重组进行数据分割，它保留了每个类的样本百分比，并将其与交叉验证相结合。当数据集对两个目标标签之一有较强不平衡性时，这会非常有用。
* http://scikit-learn.org/stable/modules/generated/sklearn.cross\_validation.StratifiedShuffleSplit.html





根据输出结果，我们得到训练F 1分数为0.8040，测试F 1分数为0.8050。

### 问题5 – 最终的F1分数

经训练与测试后的最终模型的F 1分数是什么？该分数与未经训练的模型相比如何？

### 答案：

* 最终模型的F 1分数为：
  + 训练F 1分数：0.8040
  + 测试F 1分数：0.8050
* 与未调整的模型相比，调整后模型的测试F 1得分有所增加
  + 目前高于朴素贝叶斯F 1分数