目录

[数据的预处理 1](#_Toc70869949)

[模型的选择 1](#_Toc70869950)

[超参数的调优 1](#_Toc70869951)

[模型准确率展示 2](#_Toc70869952)

[模型结果可视化 3](#_Toc70869953)

[RandomizedSearchCV+Random Forest: 3](#_Toc70869954)

[RandomizedSearchCV+GridSearchCV+Random Forest: 3](#_Toc70869955)

# 数据的预处理

使用pandas对读取数据集进行预处理；

LabelEncoder将离散型的数据转换成0到n−1之间的数。

使用平均值填充缺失值；id,gender,age,hypertension,heart\_disease,ever\_married,work\_type,Residence\_type,avg\_glucose\_level,bmi,smoking\_status为特征列；

Stroke为分类标识符；

数据极其不平衡，使用SMOTE（合成少数类过采样技术）平衡数据

# 模型的选择

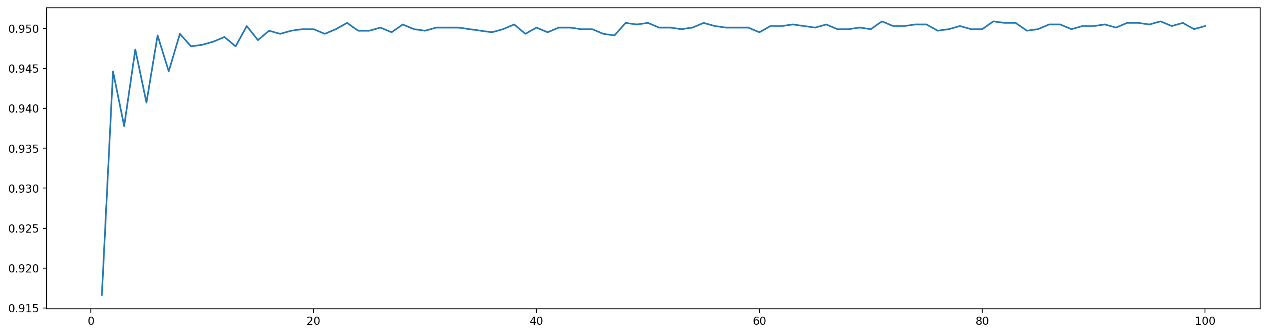
分别使用了决策树分类和随机森林的分类模型，发现随机森林的分类模型准确率相比于决策树的分类的准确率较高，数据结果如下：

|  |  |
| --- | --- |
| 模型名称 | 准确率 |
| Decision Tree | 0.9151989562948467 |
| Random Forest | 0.9510763209393346 |

# 超参数的调优

基评估器数量即：n\_estimators，这是森林中树木的数量，即基评估器的数量。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是100。

基评估器数量对随机森林的分类的模型的影响最大，我们单独画出随机森林的分类模型的学习曲线，如下图：

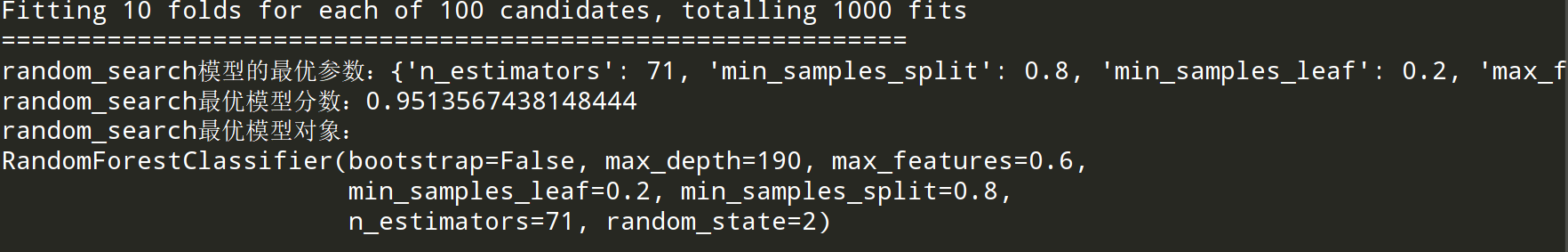


我们可以大概看出基评估器的数量在七十多时，随机森林的分类的模型的准确率达到最高，后边一直处于上下波动状态，基评估器数量越多内存消耗越大，我们应当选择基评估器相对较小，而且模型准确率较高的数值，经过计算得出的结果是71。

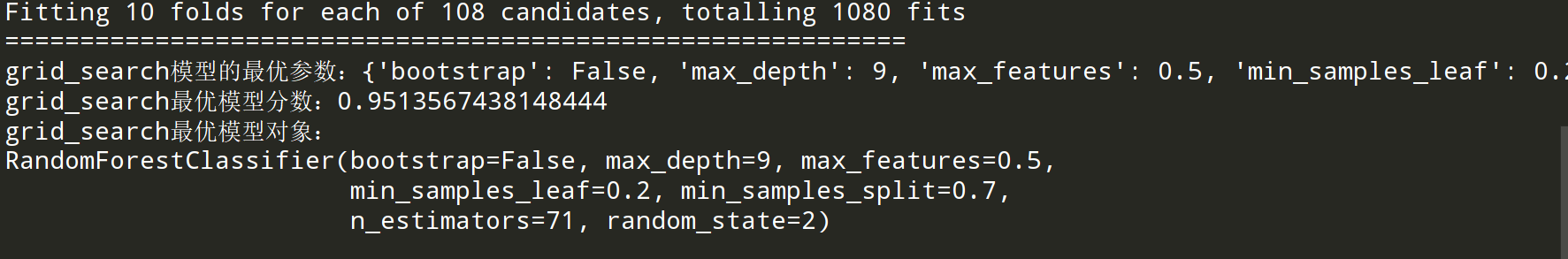
我们设定基评估器的数值为71。

GridSearchCV，它存在的意义就是自动调参，只要把参数输进去，就能给出最优化结果和参数。但是这个方法适合于小数据集，一旦数据的量级上去了，很难得到结果，我们首先使用随机搜索的算法来计算其他超参数，来缩小网格搜索的范围

其他超参数比如：bootstrap'，'max\_depth','max\_features'，'min\_samples\_leaf'，'min\_samples\_split'，对模型的准确率相对于基评估器的影响较小，计算结果如下：



因为之前模型的准确率已经相当高了，几乎差不多但是我们但是它的基评估器的数量为100，相比于71 是比较高的，我们可以选择71，而且准确率比基评估器为100的模型还略微高了一点。接下来我们利用RandomizedSearchCV 得出的结果进一步缩小范围使用GridSearchCV网格搜索法确定随机森林模型的最终参数，我们发现准确率没变。

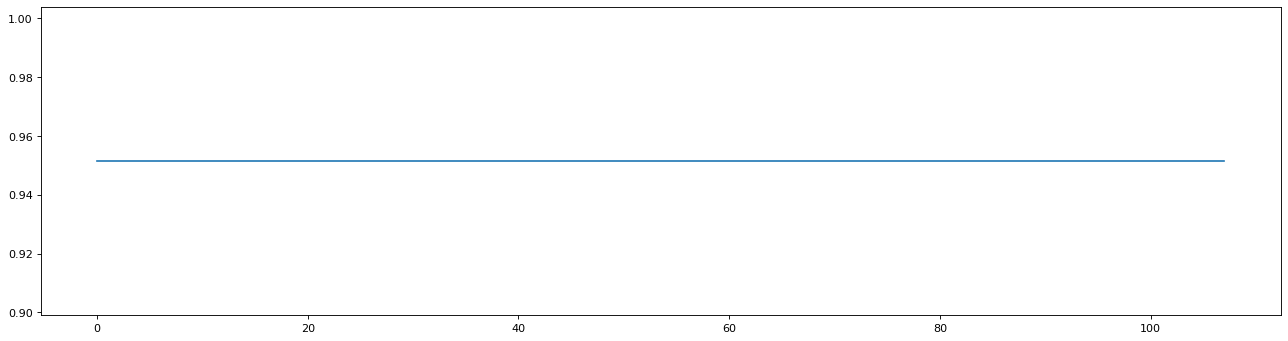


# 模型准确率展示

|  |  |
| --- | --- |
| 模型 | 准确率 |
| Decision Tree | 0.9151989562948467 |
| Random Forest | 0.9510763209393346 |
| RandomizedSearchCV+Random Forest | 0.9513567438148444 |
| RandomizedSearchCV+GridSearchCV+Random Forest | 0.9513567438148444 |

# 模型结果可视化

## RandomizedSearchCV+Random Forest:



## RandomizedSearchCV+GridSearchCV+Random Forest:

