优达学城数据分析师纳米学位项目 P5

安然提交开放式问题

说明：你可以在这里下载此文档的英文版本。

机器学习的一个重要部分就是明确你的分析过程，并有效地传达给他人。下面的问题将帮助我们理解你的决策过程及为你的项目提供反馈。请回答每个问题；每个问题的答案长度应为大概 1 到 2 段文字。如果你发现自己的答案过长，请看看是否可加以精简！

当评估员审查你的回答时，他或她将使用特定标准项清单来评估你的答案。下面是该标准的链接：评估准则。每个问题有一或多个关联的特定标准项，因此在提交答案前，请先查阅标准的相应部分。如果你的回答未满足所有标准点的期望，你将需要修改和重新提交项目。确保你的回答有足够的详细信息，使评估员能够理解你在进行数据分析时采取的每个步骤和思考过程。

提交回答后，你的导师将查看并对你的一个或多个答案提出几个更有针对性的后续问题。

我们期待看到你的项目成果！

1. 向我们总结此项目的目标以及机器学习对于实现此目标有何帮助。作为答案的部分，提供一些数据集背景信息以及这些信息如何用于回答项目问题。你在获得数据时它们是否包含任何异常值，你是如何进行处理的？【相关标准项：“数据探索”，“异常值调查”】

项目中包括了 146 位安然员工的邮件和财务特征，其中的 18 位是被人工标记成了嫌疑人，项目目标就是通过这 18 位嫌疑人的邮件和财务特征发现更多的嫌疑人。机器学习能够有效进行监督分类，根据已有的 18 位嫌疑人的信息，推测更多的嫌疑人。

异常值的识别和处理：

主要通过可视化的方式识别异常值，通过 salary 和 bonus 的散点图识别出 salary 和 bonus 中的具体异常值，其原因是来自于原始数据表格中的汇总项，这个是个明显的错误，所以直接删除掉。

对所有的财务特征进行 boxplot 可视化，发现 total_payments 以及 restricted_stock_deferred 里面含有明显的异常值，因此会这两个变量中的值进一步检查发现，total_payment 的最高者是 LAY KENNETH L，其总收入为 103559793，poi 为 True，restricted_stock_deferred 最高者是 BHATNAGAR SANJAY，值为 15456290， poi 为 False（此人的 salary 和 bonus 都是 NaN）。这些值虽然看上去比较异常，但是可能是安然高管，所以不做处理。对邮件特征进行可视化发现，KAMINSKI WINCENTY J 的 from_messages 异常地高，但是不能确定为无效的异常值。

对于数据的思考：(来自第一次 review)

- 数据集不平衡说明 accuracy 不是很好的评估指标，选择 precision 和 recall 比较合适

交叉验证时，由于数据的不平衡，选择 stratifiedshufflesplit 的方式将数据分为验证集和测试集

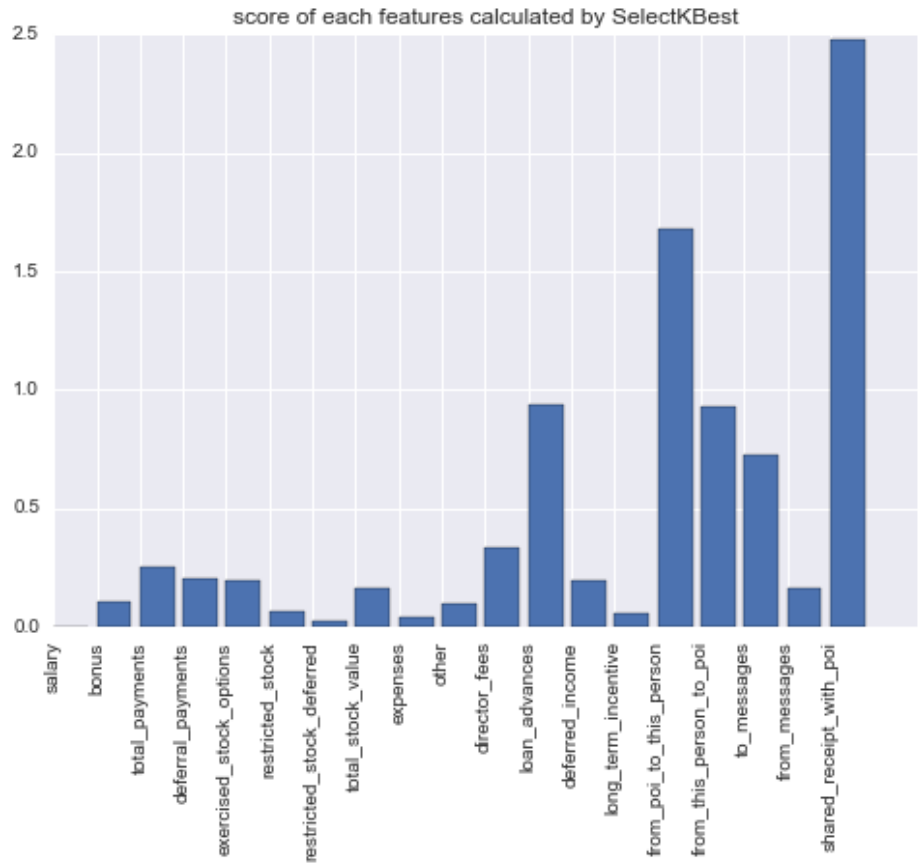
- 数据样本少，因此我们可以使用 GridSearchCV 来进行参数调整，如果较大的数据会花

很多时间，可以考虑使用 RandomizedSearchCV。

2. 你最终在你的 POI 标识符中使用了什么特征，你使用了什么筛选过程来挑选它们？你是否需要进行任何缩放？为什么？作为任务的一部分，你应该尝试设计自己的特征，而非使用数据集中现成的——解释你尝试创建的特征及其基本原理。（你不一定要在最后的分析中使用它，而只设计并测试它）。在你的特征选择步骤，如果你使用了算法（如决策树），请也给出所使用特征的特征重要性；如果你使用了自动特征选择函数（如 SelectBest），请报告特征得分及你所选的参数值的原因。【相关标准项：“创建新特征”、“适当缩放特征”、“智能选择功能”】

特征选择方法: 数据集中的每条记录都有 21 个特征（包括 poi），特征筛选过程中采用了 3 种方法

1. **直觉选择**了邮件特征中的'shared_receipt_with_poi', 'from_this_person_to_poi', 'from_poi_to_this_person'，在财务特征上选择了‘salary’‘bonus’。
2. **可视化的方法**绘制 scatterplot 观察不同特征下的嫌疑人的分布情况，发现’shared_receipt_with_poi’，‘from_messages’对嫌疑人的分类比较明显。
3. **SelectKBest** 单变量自动选择：利用 SelectKBest 对 19 个数值特征进行分数计算，并且进行可视化，结果中可以看出邮件特征的分数明显大于财务的特征分数，其中‘from_poi_to_this_person’,’from_this_person_to_poi’, ’shared_receipt_with_poi’等这些分数都非常高，其中有些结果和我的直觉推测类似，但是特征 salary 的分数有点出乎我的意料，不过最终将不会选择‘salary’这一特征。



创建新特征：主要尝试根据邮件特征建立新特征，分别使用新特征包括：
fraction_from_poi = from_poi_to_this_person/to_messages
fraction_this_person_to_poi = from_this_person_to_poi/from_messages
这两个新特征能够说明该人与 poi 之间的邮件往来与自己收发邮件数量的比例关系，理论上，嫌疑人之间一般都会有很多的邮件往来，通过计算该人与 poi 之间的邮件比例推测。

最终选择特征：(poi_id.py 的第 18 行)
['loan_advances','director_fees','shared_receipt_with_poi','from_this_person_to_poi',
'from_poi_to_this_person','bonus','deferred_income','total_stock_value','expenses']一共 9 个特征。

测试新特征：
在 poi_id.py 中的 features_list 中加入加上新特征 fraction_this_person_to_poi 之后，算法的性能从 Precision: 0.36700 Recall: 0.32700 变化到了 Precision: 0.42857 Recall: 0.36750。说明新特征的加入有助于提高对特征的识别。再加上第二个新建特征 fraction_from_poi 后，precision 和 recall 分别变成了 0.46667 和 0.4200，说明通过添加新特征能够明显改变算法的性能(关于此部分的测试可见 poi_id.py 中 123 行，测试时需要 comment 第 18 行的 final_features)

3. 你最终使用了什么算法？你还尝试了其他什么算法？不同算法之间的模型性能有何差异？【相关标准项：“选择算法”】

最终选择了决策树，另外还尝试了朴素贝叶斯，Adaboost 以及 k-近邻算法。通过对比不同算法在 training data 中的准确度以及利用交叉验证计算准确度，来对比不同算法的拟合能力以及泛化能力。
不同算法直接的差异为：
k-近邻算法的计算复杂度比较高，精度较高，本例中训练准确率 0.88，交叉验证分数 0.86
朴素贝叶斯：适合数据量少的，本例中训练准确率 0.85，交叉验证分数 0.7
决策树：计算复杂度不高，易理解，但是比较容易产生过拟合，需要调整很多参数，本例中训练准确率 0.933，交叉验证分数 0.8666
Adaboost：利用多个弱分类器的集成方法，可有效提高分类性能，泛化能力强，本例中采用单层决策树作为弱分类器，训练准确率 0.90，交叉验证分数 0.86，效果还是不错的，但是时间复杂度较高。

4. 调整算法的参数是什么意思，如果你不这样做会发生什么？你是如何调整特定算法的参数的？（一些算法没有需要调整的参数 – 如果你选择的算法是这种情况，指明并简要解释对于你最终未选择的模型或需要参数调整的不同模型，例如决策树分类器，你会怎么做）。【相关标准项：“调整算法”】

算法参数的调整表示对算法执行过程中细节进行微调，从而让算法更好地适用于特定情形。不调整参数就容易导致过拟合情况的发生（决策树中经常发生）。在对于决策树算法参数调整过程中，使用默认参数时算法准确度为 1，而交叉验证分数较低，这说明了过拟合的产生，因此，主要通过修改 max_depth 和 max_features 来调整算法性能，个人觉得这个影响决策树算法的关键参数。

	Accuracy_sco	Cross_vali	precision	recall	说明
max_depth=5 max_features = 3	0.957	0.83	0.367	0.32	较好性能
max_depth=4 max_features = 3	0.94	0.86	0.4197	0.318	性能最佳
max_depth=6 max_features = 3	0.915	0.83	0.33	0.317	性能一般
max_depth=3 max_features = 4	0.92	0.84	0.29	0.18	性能较差
max_depth=3 max_features = 4	0.92	0.83	0.26	0.12	较差

注：accuracy_sco 是直接 features 和 label 进行的测试，不是 test.py 中的 accuracy

5. 什么是验证，未正确执行情况下的典型错误是什么？你是如何验证你的分析的？【相关标准项：“验证策略”】

验证就是考察算法在不同于验证集上运行时候的分类能力，用于调整分类器的参数。验证一个很重要的任务就是将数据集分为训练和验证两个数据集，可以直接静态分割，但是这样往往会有很大问题，可以通过 K 折交叉验证得到 K 个分数，最后取平均值。未正确执行下的典型错误是不能正确描述算法的泛化能力。

我自己采用了 10 折交叉验证（见代码第 214 行），可以得到 10 个分数，同时我计算十个得分的平均值，在本例中平均分数为 0.86（还算不错）。

同时采用了 tester.py 中的函数进行验证，stratifiedshufflesplit 函数的工作原理是对数据集合进行 1000 次洗牌（n_iter = 1000）每次选取洗牌后的 0.1(test_size = 0.1)作为验证集。

6. 给出至少 2 个评估度量并说明每个的平均性能。解释对用简单的语言表明算法性能的度量的解读。【相关标准项：“评估度量的使用”】

度量：

在训练数据上的评估分数：0.92 表明训练结果良好

交叉验证评估分析：0.85 说明具有一定的泛化能力，能够适应新的数据

精确率：0.417，说明通过此算法预测的结果中，大约 41.7%是真实嫌疑人。

召回率：0.318，说明该算法可以将大约 31.8%的真实嫌疑人识别出来。

其他（留给自己思考）：

在实现算法测试过程中，在含有新特征的条件下评估得到的准确率很高，但是召回率和精准率很低，这种情况应该是发生在分类不平衡的时候，分类器对 0 的预测很好，我们把把目标只关注于 1 类的时候会发现召回率和精准率很低。发现去掉了新加的特征之后召回率和精准率有所提升。

发现调整算法的参数对 precision 以及 recall 的影响并不是很大，反而在调整特征之后会有很大的影响，那么现实机器学习过程中也会通过通过 precision 和 recall 的值来反过来调整特征吗？（应该会的吧）（比如添加了'total_stock_value'之后两者的值大幅度提升）

调整特征的顺序对 precision 以及 recall 也有影响，那么实际当中都是怎么进行排序的呢，把影响大的排在前面？

思考：cross_val_score 和 KFold 有什么区别？

验证，训练和测试之间的关系是什么。sklearn 库下面的 metrics 和 cross_validation 下面的这些函数的作用有什么大的区别吗？（难道就是验证用于测试调整参数？）

优达学城

2016 年 9 月