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

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”】

除去 TOTAL、LOCKHART EUGENE E，my_dataset 里有 144 条记录。
我的观点是，那个代表汇总数据的 TOTAL 一定要当作异常值去掉，因为没有事物会具有其他所有事物各个属性的总和，这是不成立的。至于其他人员，他可能因为在公司的地位较高而获得的报酬相应高很多这是可以理解的，这群人数量不多我觉得对整体构成的影响不是很大，所以就目前而言我不准备去掉极个别“富有的人”。 LOCKHART EUGENE E 这条记录的所有值都为 NaN，也把它删去了。

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”】

我使用了'shared_receipt_with_poi','exercised_stock_options'这两个特征（不包括后面创建的新特征），我是通过文本理解所有特征的含义，然后按照常规逻辑选择，一开始也选择过'total_stock_value'，但是后来去掉发现各指标上升了不少。我决定不进行缩放，因为我使用 MinMaxScaler 简单缩放了一下发现各指标只是在 0.1-0.2 范围波动，并没有明显的改善迹象。

我创建了新特征 **poi_email_ratio**，是先计算所有人的 from_this_person_to_poi 和

from_poi_to_this_person 的和（all_poi_email），然后用将所有人的 all_poi_email 加起来得到 total_POI_related_emails ， 再 计 算 每 个 人 的 $\frac{\textbf{all\_poi\_email}}{\boldsymbol{total\_POI\_related\_emails}}$ **（poi_email_ratio）。**

我的考虑是，from_this_person_to_poi 和 from_poi_to_this_person 两个特征的值不是很有整体特性，它只是孤零零一个值，于是我统计所有的这类值并转换成一个相对总体数据的比值，这样它便更加有整体特性，它的大小直接反映了在总体中的地位，我想这是很直观的。

['poi','shared_receipt_with_poi','exercised_stock_options']
Precision: 0.42498 Recall: 0.33000 F1: 0.37152 F2: 0.34544
Precision: 0.42234 Recall: 0.32900 F1: 0.36987 F2: 0.34421

['poi','shared_receipt_with_poi','exercised_stock_options','poi_email_ratio']
Precision: 0.43826 Recall: 0.35850 F1: 0.39439 F2: 0.37204
Precision: 0.43276 Recall: 0.35400 F1: 0.38944 F2: 0.36737

['poi','shared_receipt_with_poi','exercised_stock_options','all_poi_email']
Precision: 0.43781 Recall: 0.35200 F1: 0.39024 F2: 0.36636
Precision: 0.44266 Recall: 0.35900 F1: 0.39647 F2: 0.37310
使用 poi_email_ratio 和使用 all_poi_email 各指标均有明显提升。

由于我用了决策树分类器，这里直接使用它的属性 feature_importances_
来测试
['shared_receipt_with_poi','exercised_stock_options','all_poi_email','poi_email_ratio']
[0.59803209 0.19521269 0.19935484 0.00740038]
调换位置再测一下
['shared_receipt_with_poi','exercised_stock_options','poi_email_ratio','all_poi_email']
[0.5626457 0.23059908 0.00740038 0.19935484]

结果显示我创建的辅助特征 all_poi_email 重要性比我想要创建的最终特征 poi_email_ratio 更加重要。最终我还是将 all_poi_email 放入选择的特征列表中。

我调查了原有数据中特征值为 NaN 的情况：
{'salary': 50, 'to_messages': 58, 'deferral_payments': 106, 'total_payments': 20, 'loan_advances': 141, 'bonus': 63, 'email_address': 33, 'restricted_stock_deferred': 127, 'total_stock_value': 19, 'shared_receipt_with_poi': 58, 'long_term_incentive': 79, 'exercised_stock_options': 43, 'from_messages': 58, 'other': 52, 'from_poi_to_this_person': 58, 'from_this_person_to_poi': 58, 'poi': 0, 'deferred_income': 96, 'expenses': 50, 'restricted_stock': 35, 'director_fees': 128}

'restricted_stock_deferred、deferred_income、director_fees 含有的缺失值较多。

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”】

我最终用了 DT，我还尝试了 SVC（花的时间太久了，我变换了 C 和 kernel 这几个参数运行几次都没出结果我就关掉了，因为真的太耗时了，于是放弃）和高斯 NB（在 Recall 这一项明显低于我现在用的 DT）。

高斯 NB 的结果：

Accuracy: 0.82208 Precision: 0.44094 Recall: 0.25200 F1: 0.32071 F2: 0.27562

现在的决策树结果：

Accuracy: 0.81758 Precision: 0.44127 Recall: 0.35500 F1: 0.39346 F2: 0.36945

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”】

不同的机器学习模型因其设计原理和设计目的等不同，模型本身有不同的阈值，需要在使用的时候设定，比如在不同场所的声控开关需要设定不同音量大小的阈值，在不同水域的水坝预警系统需要设定一个与当前所在地水情相符的报警水位。同一种机器学习模型被设置了不一样的参数，就很可能会对相同的数据产生不同的效果，因此选择好机器学习的模型后应该根据项目的实际需求和数据本身对模型进行调参，使得模型更加贴合数据，使结果的指标更加优良。我是看 tester.py 生成的指标，手工调整。

如果使用 SVC 我肯定要调整一下 C 和内核，C 的话越大对训练集拟合的效果越好，但是容易过拟合，内核的选用应根据不同数据以及期望的结果来选择。

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”】

验证就是在训练好模型后对模型的性能进行评估。如果未正确执行验证，则模型可能对训练集的效果很好，但是对没有训练过的数据表现很差。如果训练集过少，则模型就不成熟，训练集过多容易造成过拟合。

在我提交的 poi_id.py 里有注释，显示了我更改 test_size 时的效果采样，当 test_size 越小，则训练集越多，模型越复杂易出现过拟合，则泛化能力减弱。而 test_size 越大，则测试集增加训练集减少，使模型欠拟合，也不利于模型的优化。

我选择的是 train_test_split 这种拆分方式，它是随机分的， test_size 这个属性代表测试集占比，random_state 为随机数种子，如果需要确保每次运行得到的随机数不变，则将其设置为

一个固定的数即可。

Tester.py 里的 StratifiedShuffleSplit 会将集合分为 n_splits 组，每组的测试集与训练集的比重可以通过 test_size、train_size 调节，参数 **random_state** 控制是将样本随机打乱，也是一个随机数种子。由于数据集很小，将其分为多个测试/训练固定比例的组进行验证有利于得到更加中肯的指标结果。

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”】

- ✓ Precision：被识别为 POI 的数据中真正为 POI 的概率。运行 4 次 tester.py 均值 0.4354125
- ✓ Recall：在所有 POI 数据中识别出 POI 的概率。运行 4 次 tester.py 均值 0.3535

Accuracy：在所有预测结果中，模型预测的正确率（POI 识别为 POI，非 POI 识别为非 POI）。运行 4 次 tester.py 均值 0.815855 【由于数据集取自复杂的现实事件环境，数据集内部不平衡，所以该指标并不是很能体现模型的性能】