

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，你是如何处理的？【相关标准项：“数据探索”，“异常值调查”】

回答如下：

项目目标——通过公开的安然财务和邮件数据集，找出有欺诈嫌疑的安然雇员

机器学习的帮助如下——通过监督式学习，学习已知特征和标签的数据，构建一个合适的算法，来预测未知雇员是否是嫌疑人

数据集探索如下：

特点	对应数值
雇员总数	146
数据集的特征总数	21
类之间的分配（POI/非 POI）	POI: 18 非 POI: 128
可用特征数	由于特征名为“email_address”的特征值均为该雇员的邮箱地址，用于分析是否为 POI 并无用处，故可去掉该特征 即有具体差异的特征有 20 个
缺失值	查看数据可知，有很多缺失值，具体体现是“NaN”

异常值调查——

打印数据类型可知：

- 1) 特征“email_address”的特征值为‘str’型，且邮件地址名称并没有具体的信息，故这里直接从‘features_list’中去掉此特征
- 2) 特征“poi”的特征值为“bool”型
- 3) 其他特征值为“int”型，但很多特征均有缺失值，表现为“NaN”，本例中，将“NaN”值处理成 0

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”】

回答如下：

- 1) 是否需要特征缩放？

——需要进行特征缩放，应该各个特征之间的差距太大了，会造成特征之间不能得到同等的重视。这里将各个特征的值范围设置在 0~1 之间。

- 2) 新特性

——在本例中，我尝试添加了新特性“poi_email_percent”，该特性计算的是，poi 与雇员之间互相发送或收取的邮件数占总邮件数的百分比

代码如下：

```
# 添加新特性：poi_email_percent
for item in my_dataset:
    temp = my_dataset[item]
    total_email = temp['to_messages'] + temp['from_messages']
    poi_email = temp['from_poi_to_this_person'] + temp['from_this_person_to_poi']

    if total_email != 0:
        temp["poi_email_percent"] = float(poi_email)/total_email
    else:
        temp["poi_email_percent"] = 0.0
#print my_dataset
features_sum.append("poi_email_percent")
```

- 3) 筛选特征：

——使用自动特征选择函数“SelectBest”进行计算时，重要性排在前五的特征以及得分如下：

重要性排名	特征	得分
1	shared_receipt_with_poi	9.0614208654849797
2	poi_email_percent	5.7578210134534
3	from_poi_to_this_person	5.5486807126564219
4	loan_advances	2.5452531997291028
5	from_this_person_to_poi	2.5146485415422259

从上面的表格可知，新特性得分为 5.7578210134534，重要性排在第二位

4) POI 标识符最终使用了什么特征？

——下表为选择特征数不同时，Accuracy, Precision, Recall, F1, F2 的取值，特征按照重要性来获取，其中，clf=tree.DecisionTreeClassifier(min_samples_split= 10)

评估\特征数	2	3	4	5	6
Accuracy	0.78844	0.82900	0.83056	0.82956	0.82933
Precision	0.09571	0.27257	0.24589	0.24811	0.24330
Recall	0.10700	0.32300	0.25400	0.26300	0.25400
F1	0.10104	0.29565	0.24988	0.25534	0.24853
F2	0.10453	0.31148	0.25233	0.25988	0.25178

从上面的表格可以看出，特征数为 3 时，Precision, Recall, F1, F2 的表现很好，而且精度 Accuracy 相差也不大，故这里最终选择 3 个特征值，即

features_list = ['poi', 'shared_receipt_with_poi', 'poi_email_percent']

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”】

回答如下：

最终使用的算法：

——最终使用的是 支持向量机 SVMs，

——还尝试了其他两种算法，朴素贝叶斯 Naïve Bayes 和 支持向量机 SVMs，其中决策树的性能最好

下表是使用的相关算法，以及参数设置

算法	参数	结果
朴素贝叶斯 MultinomialNB	alpha = 1, fit_prior=False	Accuracy: 0.37211 Precision: 0.06378 Recall: 0.34000
	alpha = 0.5, fit_prior=False	Accuracy: 0.37567 Precision: 0.06367 Recall: 0.33700
	alpha = 0, fit_prior=False	Accuracy: 0.37800

		Precision: 0.06359 Recall: 0.33500
SVMs	C=10, gamma=50	Accuracy: 0.84989 Precision: 0.33428 Recall: 0.35400 F1: 0.34386
	C=10, gamma=100	Accuracy: 0.85089 Precision: 0.36578 Recall: 0.46600 F1: 0.40985
	C=10, gamma=110	Accuracy: 0.83756 Precision: 0.31066 Recall: 0.37900 F1: 0.34144
	C=8, gamma=100 (最佳参数)	Accuracy: 0.86244 Precision: 0.40278 Recall: 0.49300 F1: 0.44335
	C=6, gamma=100	Accuracy: 0.85122 Precision: 0.34549 Recall: 0.37900 F1: 0.36147
Decision trees	min_samples_split= 25	Accuracy: 0.79644 Precision: 0.18341 Recall: 0.24100 F1: 0.20830
	min_samples_split= 20	Accuracy: 0.85067 Precision: 0.33333 Recall: 0.34400 F1: 0.33858
	min_samples_split= 15	Accuracy: 0.84567 Precision: 0.31805 Recall: 0.34000 F1: 0.32866

从上述尝试可以看出，当选择 SVMs 和 Decision trees 算法时，均可以通过调整参数，获得 0.3 以上的精确度和召回率，选择朴素贝叶斯时，达不到 0.3 的精确度。在调整的过程中，选择 SVMs 且参数设置为：C=8, gamma=100 时，可以获得最佳的性能：

Accuracy: 0.86244

Precision: 0.40278

Recall: 0.49300

F1: 0.44335

故最终选择使用 SVMs 算法。

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”】

回答如下:

调整算法的参数是什么意思?

——调整参数是指根据参数的取值不同, 算法模型会表现出不同的性能。

调整算法的参数主要是为了协调偏差和方差, 防止出现过拟合现象, 提高算法的性能——准确度。

如果你不这样做会发生什么?

——如上述使用的 3 中算法, 如果不这么调整参数, 采用默认参数, 三种算法都会出现过拟合现象, 且朴素贝叶斯和 SVMs 得到的“精确度”、“召回率”为 0, 而 Decision trees 只能得到性能很低的模型。

你是如何调整特定算法的参数的?

调整的原则是, 每次调整一个参数, 获得最好的性能后, 调整下一个参数, 直到得到最大值。

——对于朴素贝叶斯, 由于我挑选了 3 个特征, 我采取的是 MultinomialNB (多项朴素贝叶斯) 来进行分类, 在这里, 我调整了参数 alpha (拉普拉斯平滑参数), 发现调整该参数对于性能改变非常小; 同时, 由于不知道先验概率, 我调整了 fit_prior=False, 直接用一个常规概率进行计算。

——对于 SVMs, 主要是调整 C、kernel、gamma 这 3 个参数, 由于本例中使用了多个特征, 且特征量大, 最好是取 kernel 为默认值 ‘rbf’。这里, 着重调整 C 和 gamma 的值。通过尝试, 得到 C=8, gamma=100 时, 可以达到最好的性能, 具体见上述表格

——对于 Decision trees, 类似可以调整的参数有很多, 例如最大深度 max_depth, 最少样本数量 min_samples_leaf, 树叶中的最少样本数量 min_sample_leaf。这里先调整了最少样本数量 min_samples_leaf, 得到当取值 20 时, 可以达到最大值, 再调整了最大深度 max_depth, 发现性能提升并不明显。

经过各个算法对参数的调整, 发现 SVMs 获得的性能好, 故这里最终选取了 SVMs 算法。

```
from sklearn import svm
clf = svm.SVC(C=8, gamma=100)
```

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”】

回答如下:

什么是验证？

——验证是使用独立的数据集来评估分类器或回归的性能，检查是否出现过拟合现象。

未正确执行情况下的典型错误是什么？

——过拟合，

即数据在训练集上表现的精确度和召回率都很高，但是无法预测其他独立的数据集，在测试集上表现非常糟糕。

你是如何验证你的分析的？

——本例中，将数据拆分为训练集和测试集来进行验证，其中测试集占总数据的 30%，通过计算准确度，得到的值是 0.909

拆分以及验证代码如下：

```
from sklearn.cross_validation import train_test_split
features_train, features_test, labels_train, labels_test = \
    train_test_split(features, labels, test_size=0.3, random_state=42)

#print features_train
print labels_train
clf.fit(features_train, labels_train)
print clf.score(features_test, labels_test)
```

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”】

回答如下：

可采取的评估度量有：精度 Accuracy，准确度 Precision，召回率 Recall

其中 Accuracy = （正确标记的数据点数量）/（全部数据点），当出现偏斜类时，用精度衡量会不合适，例如本项目中，嫌疑人的数量远小于非嫌疑人数量，就会出现分类不均匀的情况，故需要其他的衡量指标

Precision = （正确识别的嫌疑人数量）/（正确识别嫌疑人数量+错误识别嫌疑人数量）

Recall = （正确识别的嫌疑人数量）/（嫌疑人的总数）

用上面的两个指标，不会出现偏斜的情况，更能反映算法的性能。

使用 tester.py 评估性能如下：

```
SVC(C=8, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape=None, degree=3, gamma=100, kernel='rbf',
    max_iter=-1, probability=False, random_state=None, shrinking=True,
    tol=0.001, verbose=False)
Accuracy: 0.86244 Precision: 0.40278 Recall: 0.49300 F1: 0.44335 F2: 0.47186
Total predictions: 9000 True positives: 493 False positives: 731 False negatives: 507 True negatives: 7269
```

即：SVC(C=8, gamma=100)

Precision: 0.40278, Recall: 0.49300