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

项目概述

安然曾是 2000 年美国最大的公司之一。2002 年，由于其存在大量的企业欺诈行为，这个昔日的大集团土崩瓦解。 在随后联邦进行的调查过程中，大量有代表性的保密信息进入了公众的视线，包括成千上万涉及高管的邮件和详细的财务数据。 在此项目中扮演侦探，根据安然丑闻中公开的财务和邮件数据来构建相关人士识别符。 找出有其中行为的安然员工。

我的项目主要包含以下内容：

1. 理解数据集与问题
2. 特征选择
3. 算法
4. 验证和评估

（1） 理解数据集与问题

数据集中有146个数据点（人），数据中有18个POI（person of interest嫌疑人）。使用的特征数量是20个，因为POI是我们希望预测的label标签而不是feature特征。对于该数据的分析帮助我对后续的机器学习策略给出了思路：

（1） 这个数据集不是很平衡，也就说明accuracy并不是很好的评估指标，选择precision和recall更好一些。

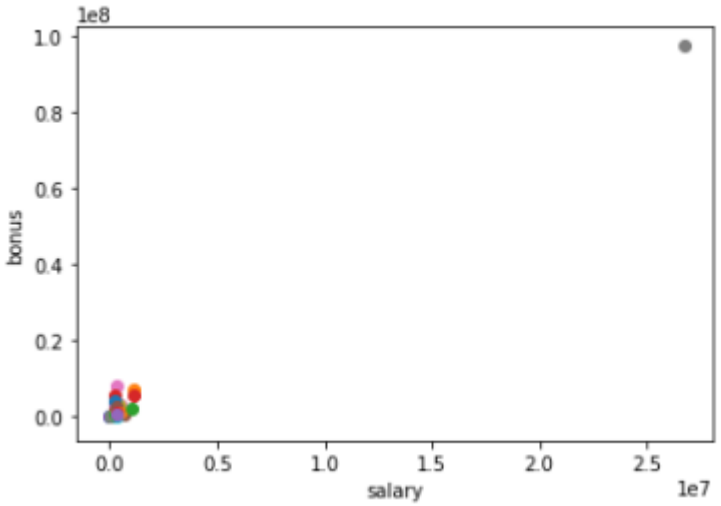
（2） 在交叉验证的时候，因为数据的不平衡，我会选用Stratified Shuffle Split的方式将数据分为验证集和测试集。

（3） 数据样本比较少，因此我们可以使用GridSearchCV来进行参数调整。如果是较大的数据则会花费较长的时间，会考虑使用RandomizedSearchCV。

查看了特征的缺失值情况：

其中”deferral_payments”,”director_fees”,”loan_advances”,”restricted_stock_deferred”有大量缺失值，特征选择步骤中会排除这几个

我在安然数据集中绘制“bonus”和“salary”的散点图，发现一个异常值。



右上角灰色点，检查发现是“TOTAL”，在数据集中将它删除。

在本数据集中，‘TRAVEL AGENCY IN THE PARK’（不是一个人）和 'LOCKHART EUGENE E'（所有特征全部为NaN，没有有用信息）都是异常值。在数据集中删掉。

查询了“salary”最高的4个人呢，其中SKILLING JEFFREY K是安然的CEO， LAY KENNETH L是安然的主席。都保留了下来，为有效数据点。

（2） 特征选择

特征选择。我先将训练集和测试集用train_test_split拆分（我设定的test_size=0.4，设定的测试集大小的不同，得分较高的特征也会有稍许改变。）。然后，用“DecisionTreeClassifier”中的“feature_importances_”挑选了重要性较大的特征。因为分离的方式是随机的，所以每次计算的特征重要性也会不同。我选取了多次得分较高的特征。最终挑选出了“salary”，“bonus”，“deferred_income”,”expenses”,”long_term_incentive”,”retricted_stock”6个特征。
使用了决策树的特征重要性进行了特征选择，特征的重要性得分如下：
Salary：0.056
Bonus:0.183
Deferred income:0.066
Expenses:0.326
Long_term_incentive:0.061
Restricted_stock:0.307

使用GaussianNB算法，用tester.py得到Accuracy=0.83,Precisn=0.42,Recall=0.34.

新特征。通过对数据的理解，构建了新特征“from_poi_to_this_person”和”from_this_person_to_poi”。以这两个为新特征作图，可得到的信息：如果发的邮件少于总数的20%是发给嫌疑人的，那么他很可能也不是嫌疑人。当我选择了“bonus”，“deferred_income”,”expenses”,“long_term_incentive”,再加两个新特征作

为feature_list。
依然使用GaussianNB算法，用tester.py得到Accuracy=0.85,Precisn=0.54,Recall=0.33.
对比得到，使用新特征，提高了最终算法的性能。

（3） 算法

我尝试了GaussianNB算法,DecisionTreeClassifier算法,SVM算法.
得到的结果如下：
GaussionNB算法，用tester.py得Accuracy=0.838,Precisn=0.421,Recall=0.348.

DecisionTreeClassifier算法,在使用选出了的特征时，效果不不是很好，recall一直小于0.3。因为决策树算法的缺点是对包含大量特征的数据容易过拟合。于是我手动精简了特征数量到3个，得到Accuracy=0.786,Precisn=0.352,Recall=0.351.过程中使用了 GridSearchCV 进行参数调整，用于系统地遍历多种参数组合，通过交叉验证确定最佳效果参数。

调整算法的意义及重要性：机器学习的目标是从训练数据中自动生成模型, 而不需要繁琐和耗时的人参与。其中一个难点是学习算法 (如决策树、聚类技术等) 要求在使用模型之前设置参数。设置这些参数的方式可以依赖于一系列因素。那就是说, 目标通常是将这些参数设置为最佳值, 使你能够以最好的方式完成学习任务。因此, 调整算法或机器学习技术, 可以简单地认为是一个优化参数来影响模型, 以使算法执行最佳的过程。调整算法参数能让我们的算法在数据集上的性能表现更好。如果不调整参数，可能会导致欠拟合或在数据集上的性能表现不佳。

决策树时手动精简到3个特征的过程：
6个特征按照“DecisionTreeClassifier”中的“feature_importances_”得出的重要性排序：

bonus	expenses	Restricted_stock	Deferred_income	Long_term_incentive	salary
1	2	3	4	5	6

特征	特征数量	Precision	Recall
1,2,3,4,5,6	6	0.359	0.244
1,2,3,4,5	5	0.357	0.247
1,2,3,4,	4	0.301	0.294
1,2,3	3	0.311	0.283
1,2,4	3	0.352	0.351

使用决策树算法，特征最终精简为[bonus,expense,deferred_income], 得到Accuracy=0.78,Precisn=0.352,Recall=0.351

SVM算法，得到的结果：Accuracy:0.712, Precision:0.263, Recall:0.565.

使用SVM算法前，用了特征缩放，把特征转化为相似的范围。Decision trees,Linear Regression不用使用特征缩放，而涉及到距离概念的SVM和K-Means clustering需要使用特征缩放。因为SVM有一条将距离最大化的分割线，如果某一点增大到其他点的

两倍，数值也扩大一倍。K-Mean clustering有一个集群中心，然后计算各数据点的距离。如果一个变量扩大一倍，数值也扩大一倍。SVM和K-Means clustering都会受到特征缩放的影响。决策树Decision trees会呈现一系列的水平线和垂线，只是在某一方向切割。在考虑某一维度时，不用考虑另一维度情况。线性回归Linear Regression，每个特征都有一个相似的系数，这个系数总与相应的特征同时出现。特征A的系数不会影响到B特征。如果把某一变量的变比例扩大一倍，特征会缩小一倍，输出没有变化。不受特征缩放的影响。

最终选择了使用GaussionNB算法。得到Accuracy=0.838,Precisn=0.421,Recall=0.348.

（4） 验证和评估

验证算法的有效性是在已知数据集上训练的模型，评估它是否能够预测未知的数据。如果不进行验证，可能发生过拟合（高方差，对数据敏感，泛化能力差）。通常将数据集分为训练集和测试集，训练集用于建立模型，测试集用于评估该模型在独立数据集上的性能，还能避免过度拟合。对于验证我使用了交叉验证cross_validation。使用的具体类型有 train_test_split（70%作为训练集，30%作为测试集） 和StratifiedShuffleSplit。

StratifiedShuffSplit的工作原理：StratifiedShuffleSplit对数据集进行分割，返回分层分裂, 即通过保留与完整集合中每个目标类相同的百分比来创建拆分。它保证取出的每个子集中的数据比例一致（POI和非POI比例一致）。StratifiedShuffleSplit的Stratified指分层，分层再这里的主要原因是避免出现poi全部被分到一边的极端情况。

Shuffle，Shuffle相当于洗牌重抽，因为是有放回抽样，我们就可以做很多很多次的迭代，（tester.py里是1000次），这样我们跑1000次，记录1000的分数，然后求平均，会使得评分更稳定，不容易受到随机性的影响（这里stratifiedshufflesplit的参数test_size默认值为0.1，也就是说每次取143*0.1 = 14个值来验证，迭代1000次之后会得到14000个预测值）。StratifiedShuffleSplit或者kfold是我们数据量不够的时候，充分利用我们的数据的好方法。

开始我使用准确率评估算法性能。但是因为安然数据集是类别失衡的，嫌疑人的数量在总人数的占比很小，所以准确率在安然数据集中不是好的评估标准。后来我使用了精准度Precisn和召回率Recall，F1。

True positive：算法预测值为1，实际值也为1（算法预测正确）

False positive：算法预测值为1，实际值为0（算法预测错误）

False negative：算法预测值为0，实际值为1（算法预测错误）

True negative：算法预测值为0，实际值为0（算法预测正确）

Precision=（True positive/Predict positive）=True positive/(True positive+False positive)

Recall=（True positive/Actual positive）=True positive/（True positive+False negative）

F1=（2*Precision*Recall）/（Precision+Recall）

以本项目为例：

精确度Precision：猜对真的嫌疑人/(猜对真的嫌疑人+误以为真的嫌疑人)，即被识

别为POI的数据中有多少是真正的POI。召回率Recall：猜对真的嫌疑人/(猜对真的嫌疑人+误以为假的嫌疑人)，即实际是POI的有多少被识别出来。

我最好的结果是使用GaussionNB算法，得到Accuracy=0.838,Precisn=0.421,Recall=0.348.

我在此确认，所提交的项目为我的工作成果，其中引用的信息出自网站、书籍、论坛、博客文章和 GitHub 代码库等”。

参考资料：

http://scikit-learn.org/stable/modules/cross_validation.html#cross-validation

http://scikit-learn.org/stable/auto_examples/model_selection/plot_precision_recall.html

优达学城论坛

http://blog.csdn.net/qq_35083093/article/details/77881677