识别安然数据集中的嫌疑人

汕头大学 陆婵芬

一.项目概述

- 本项目是利用机器学习，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。
- 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诈骗行为，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。本项目将通过机器学习提取并识别最能代表安然数据的有用特征。并且通过常用到的机器学习算法识别安然数据集中的嫌疑人，以及评估机器学习算法的性能。

二.理解数据和问题

1.理解数据集

- 该安然数据集中,一共有3066个数据点；
- 该安然数据集收集了146个安然雇员的信息，其中有18个为诈骗嫌疑人，有128个为无辜者，其中“poi”为数据集的标签。
- 该安然数据集中每个人均有20个特征信息，其中这些特征信息可以分成两大类：财务特征，邮件特征。

```
------------------number of dataset-------
the number of dataset:  3066
the number of the person: 146

------------------number of NaN detection-
number of poi:  18
number of not poi:  128
number of neither poi or not poi:  0
number of features:  20
```

安然数据集中特征列表：

```
the list of features: ['salary', 'to_messages', 'deferral_payments', 'total_payments', 'exercised_stock_optio
ns', 'bonus', 'restricted_stock', 'shared_receipt_with_poi', 'restricted_stock_deferred', 'total_stock_valu
e', 'expenses', 'loan_advances', 'from_messages', 'other', 'from_this_person_to_poi', 'director_fees', 'defer
red_income', 'long_term_incentive', 'email_address', 'from_poi_to_this_person']
```

2.探索缺失值

- 因为本数据集为真实数据集，而且该数据集为安然公司倒闭后收集的数据，所以一些数据特征可能会

存在缺失的情况。在数据集中，缺失值由“NaN”来填充。

- 在146个雇员中，每个雇员分别有20个特征。其中最多缺失值的特征为“loan_advances”,有约97%的人缺失这个特征,这个特征表示的是贷款的进展；第二大和第三大缺失值最多的特征为“director_fees”和“restricted_stock_defferred”,有大约88%的人缺失这两个特征，“director_fees”表示的是董事酬金费,“restricted_stock_defferred”表示的是限制性股票延期。而标签“poi”没有人缺失，即在该数据集中的人均确认为嫌疑人或者无辜者。当某个特征数据量缺失太多时，不利于探索该特征对数据集的贡献，所以“loan_advances”,“director_fees”,“restricted_stock_defferred”这些缺失值较多的特征需要谨慎考虑是否探究其对数据集，或者识别嫌疑人的影响。因为“loan_advance”，“director_fees”,“restricted_stock_defferred”缺失值较多，所以将这三个特征删除。
- 在该数据集中，有51个人“salary”这个特征为“NaN”，其中有一个嫌疑人的“salary”特征确认，嫌疑人的缺失占总人数的0.685%,所以这个特征可以用于探索数据集。其次，在数据集中有21人缺失“total_payment”特征，其中没有嫌疑人缺失该特征，所以该特征也可以用于探索数据集。再者有35人丢失“email_address”特征，其中没有嫌疑人缺失该特征，所以该特征也可以用于探索数据集。同样的道理，特征“bonus”嫌疑人缺失占总数的1.37%，也可以用于数据集的探索。

图2-2-1 所有特征的缺失值：

```
features that have 'NaN' and its number:
[('loan_advances', 142), ('director_fees', 129), ('restricted_stock_deferred', 128), ('deferral_payments', 10
7), ('deferred_income', 97), ('long_term_incentive', 80), ('bonus', 64), ('to_messages', 60), ('shared_receip
t_with_poi', 60), ('from_messages', 60), ('from_poi_to_this_person', 60), ('from_this_person_to_poi', 60),
('other', 53), ('salary', 51), ('expenses', 51), ('exercised_stock_options', 44), ('restricted_stock', 36),
('email_address', 35), ('total_payments', 21), ('total_stock_value', 20)]
```

图2-2-2 部分特征的缺失值：

```
number of person's salary is 'NaN': 51 ,and number of poi's salary is 'NaN': 1 the rati
o is: 0.685 %
number of person's total_payments is 'NaN': 21 ,and number of poi's total_payments is
'NaN': 0 the ratio is : 0.0 %
number of person's email_address is 'NaN': 35 ,and number of poi's email_address is 'Na
N': 0 the ratio is: 0.0 %
number of person's bonus is 'NaN': 64 ,and number of poi's bonus is 'NaN': 2 the ratio
is: 1.37 %
```

3.异常值调查

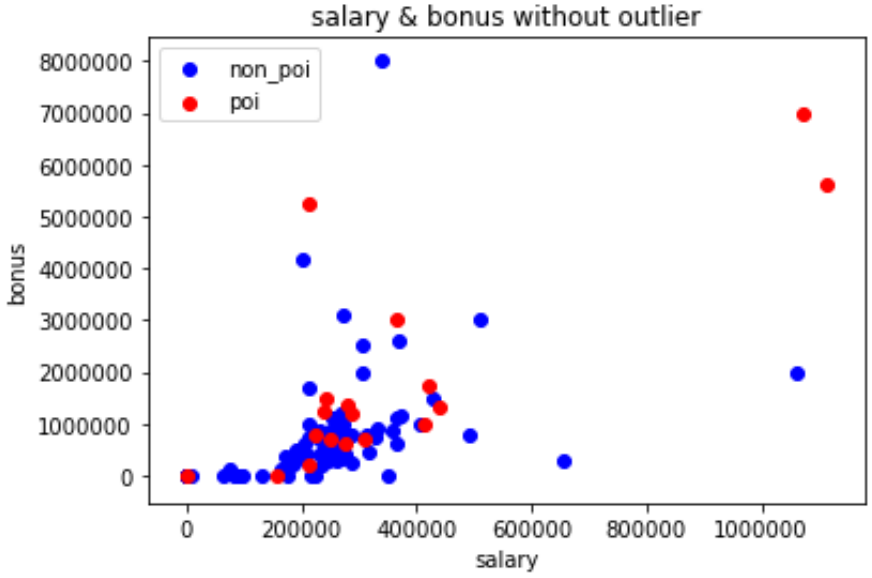
- 由于安然数据集为真实的数据集，所以可能因为传感器故障，数据输入错误或者一些反常事件导致数据集中出现异常值，所以本项目在进行数据探索前，需要先进行异常值的检测。
- 通过检查安然数据集中每个员工的名字，发现在数据集中，‘TRAVEL AGENCY IN THE PARK’并不是表示一个雇员的名称，而是公园里的旅社，所以这个值为异常值，需要删除；
- 通过探索，发现名称为'LOCKHART EUGENE E'的雇员的所有特征均为"NaN"，没有有用信息，所以可以将其删除，减少冗余。
- 本项目使用可视化的方式来判断异常值。从图2-3-1 带有异常值的数据点分布中，可知有一个点和其他的点偏离很多。通过查找文件可以这个点为总人数的特征总数，这个数据为后期计算得出，不属于真实数据集中的数据，该数据为异常值，所以可以将该数据删除。
- 当删除该异常值后，数据分布如图2-3-1 删除异常值的数据点分布图所示。该图中虽然有四个点分布比较分散，但是因为这四个点涉及嫌疑人“poi”和非嫌疑人“non_poi”，而且也有可能存在薪金和奖金较大的雇员，而这样的雇员有更大的可能会成为嫌疑人，所以这四个点不能删除。

雇员信息均缺失的列表：

```
the person  whose featuers are all  NaN: ['LOCKHART EUGENE E']
```

图2-3-1 带有异常值的数据点分布：

图2-3-2 删除异常值的数据点分布：

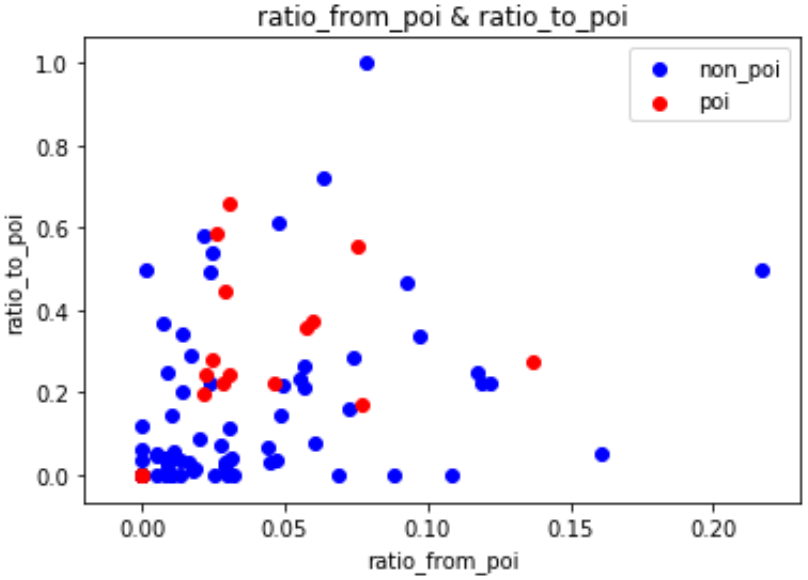


三.优化特征选择

1.创建新特征

因为该数据集主要是从邮件中抽出信息的，而且嫌疑人之间的邮件可能会更频繁，所以本项目创建了两个新特征：“ratio_from_poi”和“ratio_to_poi”,其中“ratio_from_poi”表示的是来自嫌疑人的邮件占该人收到总邮件的比例，“ratio_to_poi”表示的是该人发送给嫌疑人的邮件占该人发送总邮件的比例。从图3-1-1 “ratio_from_poi & ratio_to_poi”图可知，嫌疑人的数据点和非嫌疑人数据点分布较明显。

图3-1-1 来自嫌疑人邮件比例和给嫌疑人邮件比例数据分布图：



根据选择和调整算法，最终选用朴素贝叶斯算法。利用最终算法对新特征进行评估。如图3-2-1 最终算法利用交叉验证和分类报告中可知，包含新建特征后，算法的精确率，召回率，F1值与不含新建特征的精确率，召回率，F1值相差不大。同时利用test.py进行验证，包含新建特征后，算法精确率提高了约0.07，而召回率，F值和F2值均有所下降。在未包含新建特征的预测中，被识别为POI的数据中有761个是真正的POI，有796个为错误识别为POI。在包含新建特征的预测中，被识别为POI的数据中有653个为真正的POI，有665个为错误识别为POI。综上，因为新建特征对最终算法的泛化能力影响不大，但是添加新特征会增加运行时间，所以在最终算法中，不将新建特征添加到features_list中。

图3-1-2 最终算法得分（未包含新特征和包含新特征）：

```
------------------------classifiers-----------------------------------------------------
the best estimator: GaussianNB(priors=None)
the best score: 0.847
---------------------------------------
new features with best estimator: GaussianNB(priors=None)
new features with best score: 0.867
detailed classification report:
             precision    recall  f1-score   support

        0.0       0.91      0.92      0.91      3843
        1.0       0.38      0.34      0.36       557

avg / total       0.84      0.85      0.84      4400

-------------------------------------------------------------------------------------------
detailed classification report with new features:
             precision    recall  f1-score   support

        0.0       0.91      0.96      0.93      3500
        1.0       0.51      0.30      0.38       500

avg / total       0.86      0.88      0.86      4000
```

图3-1-3 最终算法性能（未包含新创建的特征）：

```
GridSearchCV(cv=None, error_score='raise', estimator=GaussianNB(priors=None),
       fit_params=None, iid=True, n_jobs=1, param_grid={},
       pre_dispatch='2*n_jobs', refit=True, return_train_score='warn',
       scoring=None, verbose=0)
        Accuracy: 0.85464       Precision: 0.48876      Recall: 0.38050 F1: 0.42789     F2: 0.39814
        Total predictions: 14000        True positives:  761    False positives:  796   False negatives: 1239
True negatives: 11204
```

图3-1-4 最终算法性能（包含新创建的特征）：

```
GridSearchCV(cv=None, error_score='raise', estimator=GaussianNB(priors=None),
       fit_params=None, iid=True, n_jobs=1, param_grid={},
       pre_dispatch='2*n_jobs', refit=True, return_train_score='warn',
       scoring=None, verbose=0)
        Accuracy: 0.85629       Precision: 0.49545      Recall: 0.32650 F1: 0.39361     F2: 0.35040
        Total predictions: 14000        True positives:  653    False positives:  665   False negatives: 1347
True negatives: 11335
```

2.选择特征

因为一共有21个特征标签，但是不是全部的特征对本项目识别嫌疑人有很大的贡献，所以本项目中将选取对识别嫌疑人有较多贡献的特征。同时能使模型泛化能力更强，减少过拟合现象出现。本项目中使用单因素方差分析法，通过feature_selection库的SelectKBest算法，得出部分特征标签的F值得分，基于统计检验得出的F值得分选择最佳特征。F值表示整个拟合方程的显著，F值越大，表示方程越显著，拟合程度也越好。如下图3-2-1为部分特征列表（除“email_address”特征）的得分从高到低的排序。在这19个特征中，其三个得分最高的特征为“exercised_stock_options”,"total_stock_value"和“bonus”。其中“exercised_stock_options”的得分为25.0.98，紧接着为“total_stock_value”的得分为24.468，说明在本数据现有19个特征中，这三个特征与“poi”的相关性最强，接着“salary”与“poi”的得分约为18.576。因为特征的得分越大，表示该特征与标签POI的相关性更大。所以本项目根据SelectKBest通过拟合后的得分大于10的特征的5个最佳特征。

图3-2-1 SelectKBest的特征的F值得分：

```
--------------------SelectKBest features scores------------
('exercised_stock_options', 25.098)
('total_stock_value', 24.468)
('bonus', 21.06)
('salary', 18.576)
('deferred_income', 11.596)
('long_term_incentive', 10.072)
('restricted_stock', 9.347)
('total_payments', 8.867)
('shared_receipt_with_poi', 8.746)
('loan_advances', 7.243)
('expenses', 6.234)
('from_poi_to_this_person', 5.345)
('other', 4.205)
('from_this_person_to_poi', 2.427)
('director_fees', 2.108)
('to_messages', 1.699)
('deferral_payments', 0.217)
('from_messages', 0.164)
('restricted_stock_deferred', 0.065)
----------------------------------------------------------------------
```

3.特征缩放

特征缩放是用来标准化数据特征的范围。特征缩放能使机器学习算法工作更好,提高模型收敛速度，提高模型精度。在多特征评价的体系中，因为各个特征指标的性质不同，通常具有不同的量纲和数量级。当各个特征指标的水平相差很大时，如果直接用原始指标值进行分析，就会突出数值较高的指标在综合分析中的作用，相对削弱数值水平较低指标的作用。因此，为了保证结果的可靠性，需要对原始指标数据进行标准化处理。其中决策树和随机森林是不需要特征缩放的算法，而SVM和K-Mean聚类是需要进行特征缩放的算法。因为本项目将尝试朴素贝叶斯算法和决策树算法，所以不需要进行特征缩放。

四.选择和调整算法

1.选择算法

- 因为本项目数据集数据量不大，而且数据集中有特征和标签分类，所以这里使用监督学习的算法对本项目数据进行探索。这里我选择利用朴素贝叶斯的高斯算法和决策树算法对数据集进行探索，并对他们的性能进行比较，选取性能较高的一个。
- 因为本项目的数据样本比较少，因此我们可以使用GridSearchCV来进行参数调整，提高工作速度。同时，在交叉验证的时候，因为数据的不平衡性，我们会选用Stratified Shuffle Split的方式将数据分为验证集和测试集，利用验证集调整参数，利用测试集测试模型的泛化能力。

2.验证算法的重要性

- 在机器学习中，我们经常会将数据集分为训练集跟测试集这两个子集，前者用以建立模型，后者则用来评估该模型对未知样本进行预测时的精确度，正规的说法是泛化能力。我们需要在测试集上进行验证，来确定训练集是否“过拟合”。
- 当进行算法选择时，我们需要一些指标来确定或者衡量所选算法是否合理。即我们需要进行算法的验证，验证算法性能是否最优，或者算法是否能运用在实际中，即算法的泛化能力是否足够的大。本项目利用交叉验证的方式，使用cross_validation库的StratifiedShuffleSplit算法对数据集进行分离，分离成验证集和测试集。其中训练集用于拟合模型，通过设置分类器的参数，训练分类模型。而测试集可以用于评估模型的精确度（即泛化能力）。因为本项目的数据很不平衡，说明accuracy不是很好的评估指标，所以本项目使用metrics库的classification_report算法得到的精确率，召回率和F1得分来评估算法拟合模型的能力。其中当精确率，召回率和F1得分越高时，表明算法性能更优。
- 在交叉验证的时候，因为数据的不平衡性，如果选用train_test_split算法，很可能会出现训练集中集中分布非POI或者集中分布POI，这样会导致模型的泛化能力较低。所以在本项目使用Stratified Shuffle Split的方式分离数据集。Stratified Shuffle Split交叉验证使用迭代器，通过保持每类样品的百分比来产生分成，并且在每层上随机产生验证集或者训练集的下标。这样的方式可以尽可能使POI和非POI随机分布在验证集和测试集。

3.决策树算法探索

- 因为一般算法中均存在一些参数，而不同的参数值选择会影响最终算法的性能的好坏。比如，SVC算法中的“C”参数控制决策边缘的光滑程度和包含训练数据点的数量。当C越大时，算法产生的决策边缘会更弯曲，同时也包含更多的正确的训练点。而当C越小时，算法包含的训练点会较少，这样会导致预测时更多的不准确性。所以我们有必要对算法的参数进行调整，从而使算法能获得更好的性能。
- 在本项目使用tree库的DecisionTreeClassifier算法，并且调整“min_sample_split”,“min_sample_leaf”参数的值。“min_sample_split”参数表示分割内部节点所需要的最小的样本数，恰当地分割内部节点最小样本数，可以提高精确率和避免过拟合。“min_sample_leaf”表示在叶节点上所需的最小样本数，恰当的叶节点上最小样本数可以使算法的性能有所提升。
- 因为数据集数量不大，所以利用GridSearchCV调整参数。如下图为通过GridSearchCV调整后得到的最佳模型。利用得到的模型来判断4400个预测样本，结果如图4-1-1 决策树参数调整后性能。利用classification_report进行判断，得到的精确率和召回率均为1，可知该模型过拟合。在利用test.py更多的预测样本来验证模型，结果如图4-1-2所示。由图4-1-2可知，在被识别为POI的数据中，有539为真正的POI，但是有1389为非POI。在被识别为非POI的数据中，有10611个为非POI，有1461为真正的POI，可知，该模型的泛化能力不高，其精确率仅约为0.28，而召回率约为0.27。

通过GridSearchCV调整参数，得到的最佳模型：

```
the best estimator: DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=None,
            max_features=None, max_leaf_nodes=None,
            min_impurity_decrease=0.0, min_impurity_split=None,
            min_samples_leaf=1, min_samples_split=2,
            min_weight_fraction_leaf=0.0, presort=False, random_state=None,
            splitter='best')
the best score: 1.0
```

图4-1-1 决策树参数调整后性能：

```
detailed classification report:
             precision    recall  f1-score   support

        0.0       1.00      1.00      1.00      3843
        1.0       1.00      1.00      1.00       557

avg / total       1.00      1.00      1.00      4400
```

图4-1-2 运行test.py，决策数算法性能验证：

```
GridSearchCV(cv=None, error_score='raise',
       estimator=DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=None,
            max_features=None, max_leaf_nodes=None,
            min_impurity_decrease=0.0, min_impurity_split=None,
            min_samples_leaf=1, min_samples_split=2,
            min_weight_fraction_leaf=0.0, presort=False, random_state=None,
            splitter='best'),
       fit_params=None, iid=True, n_jobs=1, param_grid={},
       pre_dispatch='2*n_jobs', refit=True, return_train_score='warn',
       scoring=None, verbose=0)
        Accuracy: 0.79643       Precision: 0.27956      Recall: 0.26950 F1: 0.27444     F2: 0.27145
        Total predictions: 14000        True positives:  539    False positives: 1389   False negatives: 1461   True negatives: 10611
```

4.朴素贝叶斯的高斯算法探索

本项目在探索另一个监督学习的算法：朴素贝叶斯的高斯算法。因为这个算法没有参数可以调整，所以直接利用算法进行探索。利用test.py进行验证，得到的精确率为0.48，召回率约为0.38,F1值约为0.43。在14000个预测样本中，被识别为POI的数据中有761个是真正的POI，有796个为错误识别的POI。同时，在本项目中，利用4400个预测样本进行预测，平均精确率为0.84，其中有38%概率能正确预测真正的POI。而有91%的概率能正确预测不是POI的雇员。

图4-1-3 朴素贝叶斯高斯算法性能，test.py结果：

```
GridSearchCV(cv=None, error_score='raise', estimator=GaussianNB(priors=None),
       fit_params=None, iid=True, n_jobs=1, param_grid={},
       pre_dispatch='2*n_jobs', refit=True, return_train_score='warn',
       scoring=None, verbose=0)
        Accuracy: 0.85464       Precision: 0.48876      Recall: 0.38050 F1: 0.42789
F2: 0.39814
        Total predictions: 14000        True positives:  761    False positives:  796
False negatives: 1239   True negatives: 11204
```

图4-1-4 朴素贝叶斯高斯算法验证性能：

```
new features with best estimator: GaussianNB(priors=None)
new features with best score: 0.867
detailed classification report:
             precision    recall  f1-score   support

        0.0       0.91      0.92      0.91      3843
        1.0       0.38      0.34      0.36       557

avg / total       0.84      0.85      0.84      4400
```

3.最终算法选择和总结

3.最终算法选择和总结

- 由上面决策树算法和朴素贝叶斯算法性能的分析可知，利用Stratified Shuffle Split的方式分离数据集，进行算法建模，利用classification_report进行算法性能验证，得到的结果比较可知，在这样的前提下，决策树建立的模型比朴素贝叶斯的高斯算法建立的模型更适合这个数据集。所以最终的算法选择朴素贝叶斯的高斯算法。
- 本项目首先对数据集进行初步的探索，了解数据集的数量，特征,异常值等信息，其次创建新特征ratio_to_poi和ratio_from_poi，并且了解到它对最终算法的贡献不大，所以最终特征中并未包含它。本项目利用SelectKBest进行单变量特征选择，利用得分最终确认了5特征：“salary”,“bonus”，“exercised_stock_options”,“deferred_income”，“total_stock_value”。最后，通过对决策树和朴素贝叶斯的高斯算法进行建模，并且比较它们的建模能力，最终确认使用决策树算法进行建模，来识别安然数据集中的嫌疑人。