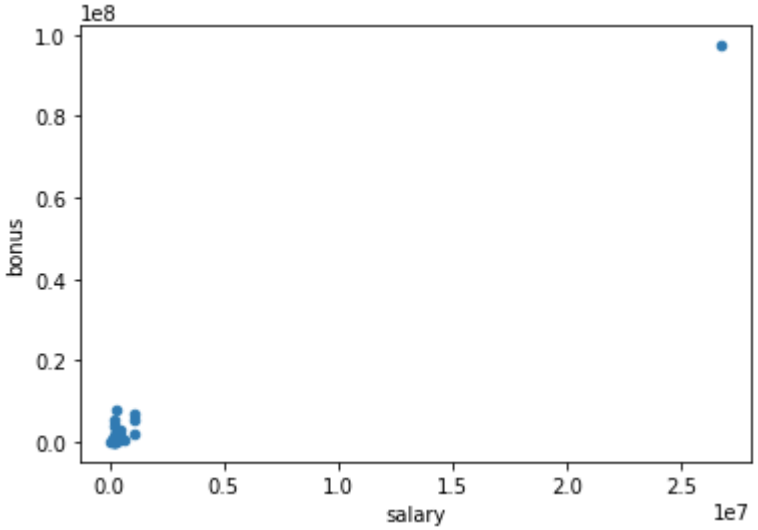
优达学城数据分析师纳米学位项目 P5

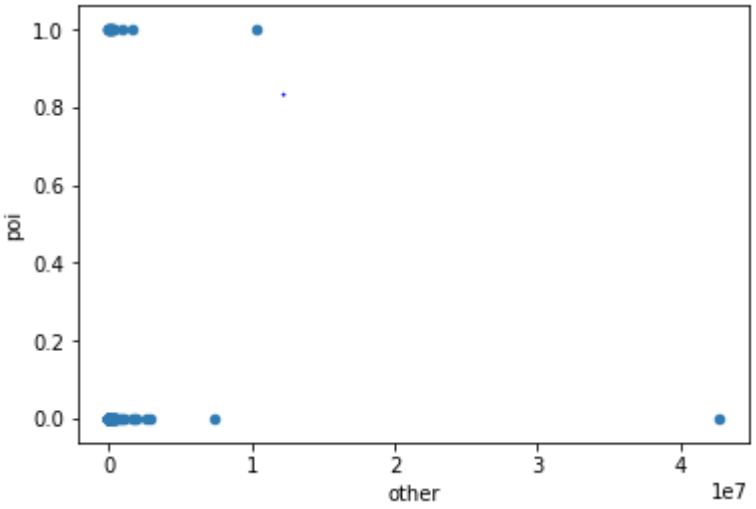
安然提交开放式问题

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”】

   本项目目标是根据公开的安然财务和电子邮件数据集，构建机器学习算法，找出有欺诈嫌疑的安然员工，我使用的数据是 final_project_dataset.pkl，它是处理过的 E+F 数据，机器学习将帮助我们从已用的数据集里生成模型然后对新数据集做预测，得出谁是嫌疑人的结果，最后对结果进行评估。
   该数据集有 146 条目标数据，有 20 个特征，1 个标签结果。选定其中的几个名称和结果相关性可能较高的特征做可视化异常值分析。
   对 bonus 和 salary 做可视化观察发现一个 TOTAL 的异常值



   通过对 other 和 poi 作可视化观察发现同样有个叫 TOTAL 的异常值



输出该目标查看各个特征

	bonus	deferral_payments	deferred_income	director_fees	email_address	exercised_stock_options	expenses	from_messages	from_poi_to_this_person	from_this_person_to_poi	...	long_term_incentive	other	poi	restricted_stock	restricted_stock_deferred	salary	shared_receipt_with_poi	to_messages	total_payments	total_stock_value
TOTAL	97343619.0	32083396.0	-27992891.0	1398517.0	NaN	311764000.0	5235198.0	NaN	NaN	NaN	...	48521928.0	42667589.0	False	130322299.0	-7576788.0	26704229.0	NaN	NaN	309886585.0	434509511.0

（图片可放大查看）
再通过字面分析应该是指各特征的各自所有目标的总和，应该是处理数据的时候携带了电子表格所统计的值，所以应该要把这个 TOTAL 目标给去掉。
再根据字面意思 THE TRAVEL AGENCY IN THE PARK 不是一个人名字也去掉
LOCKHART EUGENE E 所有值为空也删去。

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”】

   总体上说改数据集缺失的比较严重，除了最后的训练结果 poi 以外的所有特征没都含有缺失值。

   我这里需要剔除的特征，缺失超过一半的特征我就先删除了，该数据集的总目标数是 146 也就是低于 73 的我就直接排除了。

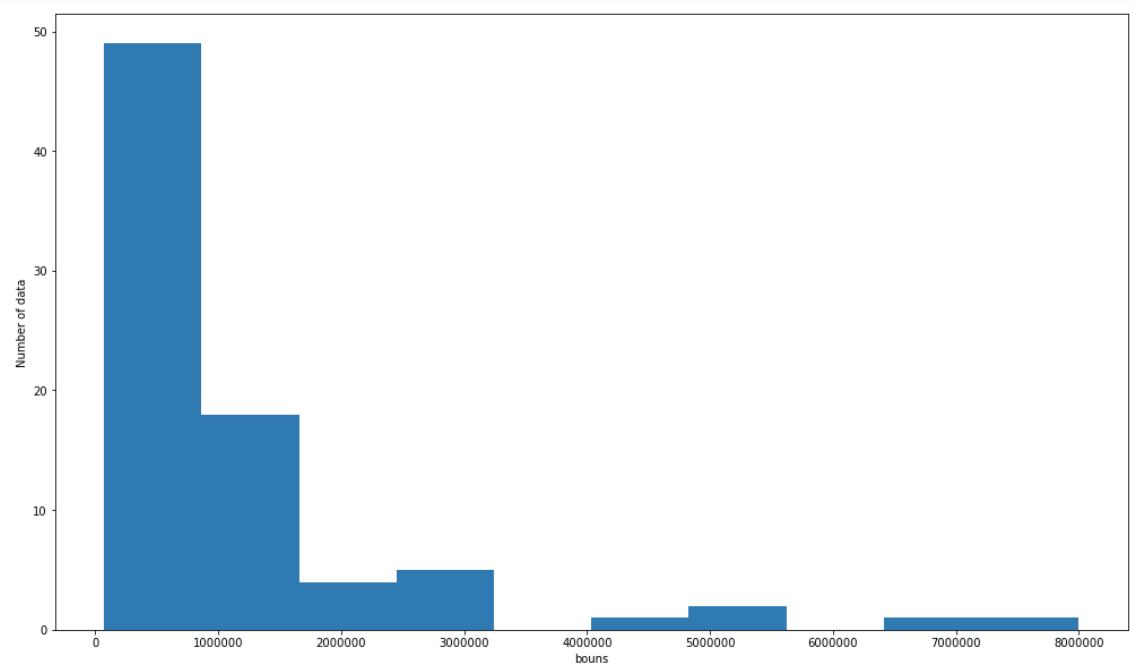
   缺失超过一般的特征由于数量不够而且部分数据比如 restricted_stock_deferred 的对象里完全没有 poi，因此容易对最后的结果造成很大的误导故不选作为训练特征。
   保留下的特征有

   email_address 111 non-null object
   exercised_stock_options 102 non-null float64
   expenses 95 non-null float64
   from_messages 86 non-null float64

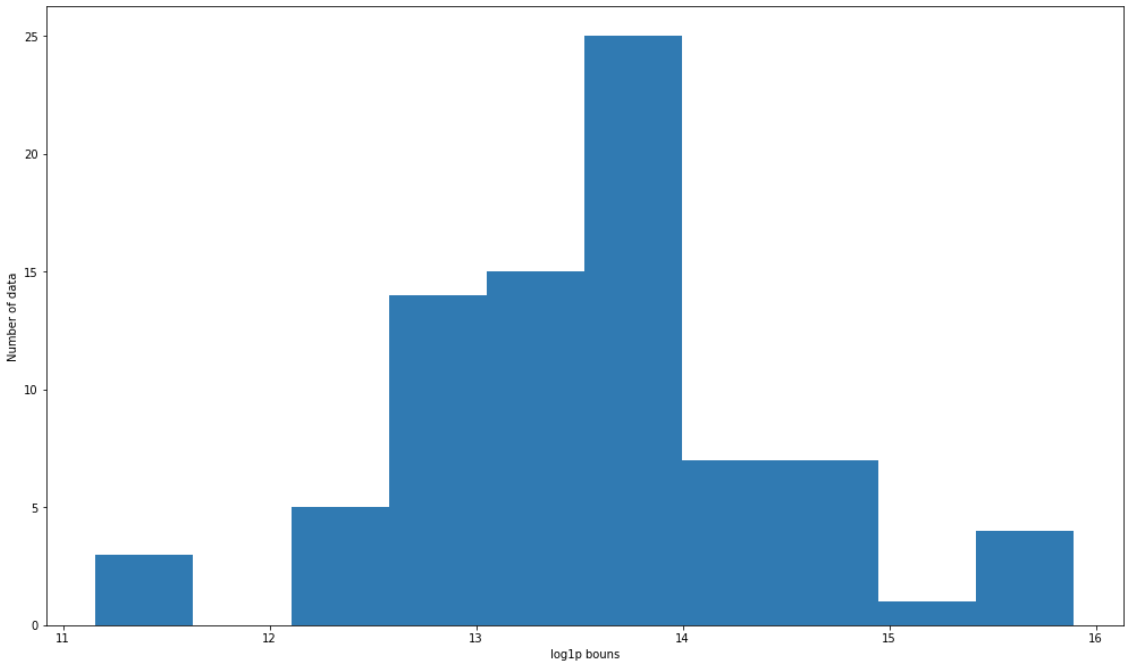
from_poi_to_this_person 86 non-null float64
from_this_person_to_poi 86 non-null float64
other 93 non-null float64
restricted_stock 110 non-null float64
salary 95 non-null float64
shared_receipt_with_poi 86 non-null float64
to_messages 86 non-null float64
total_payments 125 non-null float64
total_stock_value 126 non-null float64

接着移除 email_address 原因是与结果无关， 剔除的特征有
email_address 111 non-null object
deferral_payments 39 non-null float64
deferred_income 49 non-null float64
director_fees 17 non-null float64
loan_advances 4 non-null float64
long_term_incentive 66 non-null float64
restricted_stock_deferred 18 non-null float64

接下来我再在这些保留的特征里进行建立新的特征和选择原有的特征作为训练特征
新建特征，观察邮件的首发情况将
'from_this_person_to_poi','from_messages','from_poi_to_this_person','to_messages'这四个数量的特征转化为相关的与 poi 接收和发送率即 from_ratio 和 to_ratio。将没有值的目标用平均数来补充，最后去掉这四个特征。
特征缩放，观察其余的各个留存的特征均为数值型，但是数值之间的差距巨大很多不在同一范围内，我将逐一的来查看各个特征
bonus 斜态分布，且数值相差巨大大部分都集中在 1000000 以下，我选择做一个对数转化同时将其转化为正态分布，这样的好处是，它可以额加快梯度下降求最优解的速 度，还可以提高计算的精度
之前的图形



使用对数之后的图形



将其做对数处理后赋值并对其缺失值补充平均数

类似的特征还有 exercised_stock_options，expenses，other，shared_receipt_with_poi，total_payments， salary 都一并采用类似的方式处理。

restricted_stock 特征很奇怪有一个很大的负数值这个特征一般不会是负数，有可能是失误，我把它处理为正然后用对数处理赋值。

total_stock_value 特征也有一个负数，stock_value 不可能为负数，所以做绝对值处理。然后再用对数做处理。

最后的结果如图

```
person_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 145 entries, ALLEN PHILLIP K to YEAP SOON
Data columns (total 12 columns):
bonus                        145 non-null float64
exercised_stock_options      145 non-null float64
expenses                     145 non-null float64
other                        145 non-null float64
poi                          145 non-null bool
restricted_stock             145 non-null float64
salary                       145 non-null float64
shared_receipt_with_poi      145 non-null float64
total_payments               145 non-null float64
total_stock_value            145 non-null float64
from_ratio                   145 non-null float64
to_ratio                     145 non-null float64
dtypes: bool(1), float64(11)
memory usage: 18.7+ KB
```

对于新特征和原有特征使用决策树算法做了对比，

原有特征的测试结果如下

```
        Accuracy: 0.79667       Precision: 0.24589      Recall: 0.25400 F1: 0.24988     F2: 0.25233
        Total predictions: 15000        True positives:  508    False positives: 1558   False negatives: 1492
 True negatives: 11442
```

使用了新特征的测试结果如下

```
        Accuracy: 0.84133       Precision: 0.40750      Recall: 0.41850 F1: 0.41293     F2: 0.41625
        Total predictions: 15000        True positives:  837    False positives: 1217   False negatives: 1163
 True negatives: 11783
```

可以看出新特征的结果要好很多。

之后我在此基础上做了特征选择使用的是针对树优化的 SelectFromModel，提出重要性超过 1.25 倍的均值的特征

最后选出三个分别是 exercised_stock_options,shared_receipt_with_poi,from_ratio。

然后用这个新的选择特征集做决策树的验证最后的结果是 Precision:0.49598，Recall:0.37050 详细如下：

```
        Accuracy: 0.86587       Precision: 0.49598      Recall: 0.37050 F1: 0.42416     F2: 0.39025
        Total predictions: 15000        True positives:  741    False positives:  753   False negatives: 1259
 True negatives: 12247
```

总体计算结果稍稍高了些，而且简化了特征只有三个特征，之前是 11 个特征，缩小了计算成本。

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”】

我尝试了 DecisionTree，LinearSVC，GaussianNB，RandomForest，特征是使用了表现更好的新特征

DecisionTree 训练用时 1.89 秒，结果如下

```
    Accuracy: 0.83993       Precision: 0.40281      Recall: 0.41550 F1: 0.40906     F2: 0.41290
    Total predictions: 15000        True positives:  831    False positives: 1232   False negatives: 1169
 True negatives: 11768
```

LinearSVC 训练用 13.4 秒，结果如下

```
        Accuracy: 0.77473       Precision: 0.15679      Recall: 0.15750 F1: 0.15715     F2: 0.15736
        Total predictions: 15000        True positives:  315    False positives: 1694   False negatives: 1685
True negatives: 11306
```

GaussianNB 训练用时 1.78 秒，结果如下

```
        Accuracy: 0.81940       Precision: 0.34934      Recall: 0.41100 F1: 0.37767     F2: 0.39699
        Total predictions: 15000        True positives:  822    False positives: 1531   False negatives: 1178
True negatives: 11469
```

RandomForest 训练用时 42 秒，结果如下

```
        Accuracy: 0.85873       Precision: 0.41439      Recall: 0.14400 F1: 0.21373     F2: 0.16561
        Total predictions: 15000        True positives:  288    False positives:  407   False negatives: 1712
True negatives: 12593
```

综上我选择了表现最好的 DecisionTree 算法，时间用的少，结果也很不错，精确率和召回率都达到了 0.4 以上。

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”】

我对选择的表现最好的 DecisionTree 算法做参数优化，我选择的使用 GridSearch 来寻找最佳的参数。通过设置可能的 min_samples_leaf 和 min_samples_split 值来便利选出最佳的参数再建模。由于我设立标准是 precision 和 recall 尽可能的高分所以我用 F1 分数作为训练标准，f1 分数为 F1 = 2 * (precision * recall) / (precision + recall)。

最后我的分数是 Precision:0.45357,Recall:0.39600

详细如下：

```
        Accuracy: 0.85520       Precision: 0.45103      Recall: 0.39600 F1: 0.42173     F2: 0.40590
        Total predictions: 15000        True positives:  792    False positives:  964   False negatives: 1208
True negatives: 12036
```

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”】

验证就是对我的模型经行评分,如果没有正确执行的话模型有可能出现过拟合,我使用的是 k 折交叉验证通过将训练集分成 K 份,取第 K 份作为验证集,其余的作为训练集来训练我的模型，然后通过 K 次重复，每个样本都验证过一次，通过平均或其他方式找出唯一的最佳的参数组合,最后的结果是 min_ssamples_split:2,min_samples_leaf:3

最后的结果是 Precision:0.43054,recall:0.3950

详细如下：

```
        Accuracy: 0.84987       Precision: 0.43054      Recall: 0.39050 F1: 0.40954     F2: 0.39790
        Total predictions: 15000        True positives:  781    False positives: 1033   False negatives: 1219
True negatives: 11967
```

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”】

由于该数据集分布并不平衡，比如财务相关数据总有少部分高层的值特别高，所以采用

accuracy 并不适合，accuracy 更适合那种平均分布的数据集，因此设定标准是 precision 和 recall，Precision 是查准率具体是指正确的目标/总的目标数，Recall 是指查全率，具体是指正确的目标数/样本里的目标数。还可以通过 F1 来对 Precision 和 Recall 做整体评估。