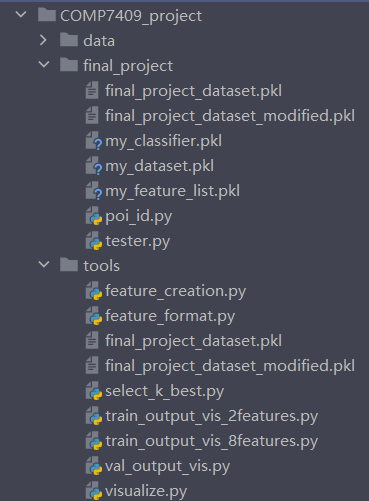
## 1.代码结构



poi\_id.py：主文件。

运行最终特征选择、特征缩放、各种分类器（可选）及其结果。

最后，转储分类器、数据集和特征列表，以便任何人都可以检查结果。

tester.py：用于验证和评估分类器、dumping 和 loading pickle 文件的功能。

/tools/z

feature\_format.py：

将数据从字典格式转换为 numpy 数组并将目标标签与特征分开以使其适合机器学习过程的函数。

feature\_creation.py

-用于创建两个新功能的函数 - “poi\_email\_ratio”和“exercised\_stock\_options”。

select\_k\_best.py

-使用 Sci-kit Learn 的 SelectKBest 选择 k 个最佳特征的功能，按分数降序排序。

visualize.py

-用于绘制由 POI 和非 POI 着色的任意两个要素的绘图的功能

**新增的py文件都是我画图写的脚本，这个环境要求不高，本地都可以跑**

**install env 之后 直接运行 poi\_id.py 就可以.**

## 2.初看数据集

总共有21个特征：

['salary', 'to\_messages', 'deferral\_payments', 'total\_payments', 'exercised\_stock\_options', 'bonus', 'restricted\_stock', 'shared\_receipt\_with\_poi', 'restricted\_stock\_deferred', 'total\_stock\_value', 'expenses', 'loan\_advances', 'from\_messages', 'other', 'from\_this\_person\_to\_poi', **'poi'**, 'director\_fees', 'deferred\_income', 'long\_term\_incentive', 'email\_address', 'from\_poi\_to\_this\_person']

布尔值 poi 表示此人是否是感兴趣的人

该项目的目的是找到将这些人与其他人区分开来的区别特征。

**可视化数据**

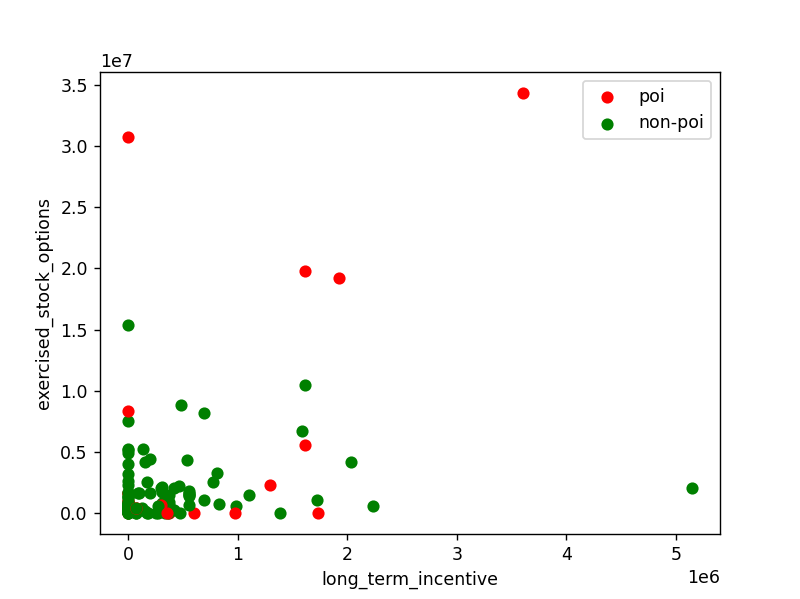
简单地创建任意两个特征的图并将它们可视化，寻找趋势并大致了解数据的变化

仅仅展示数据，大概看一下数据，对后面特征选择没啥帮助

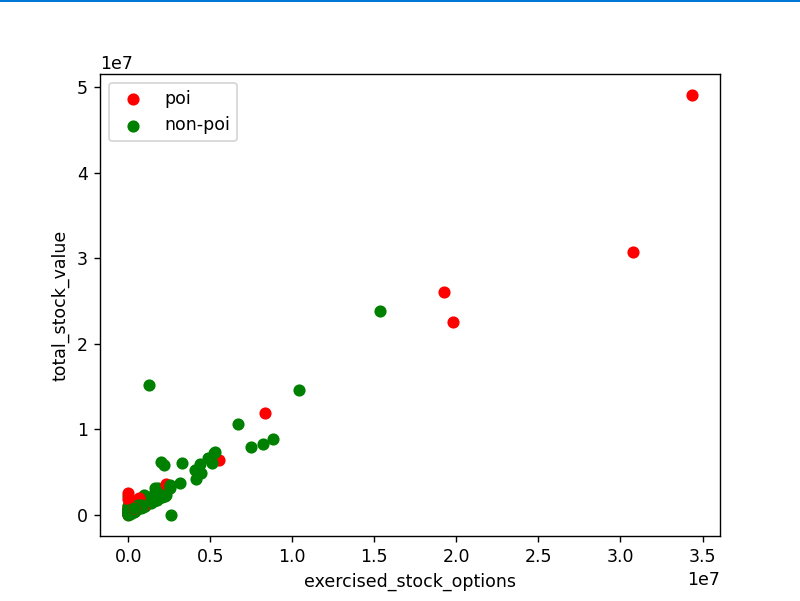
**如何生成：依次注释掉第178-183的代码，每一行都代表一个图**

这四幅图选择性地放，因为后面有热力图

1. Exercised Stock Options vs Long-term Incentive

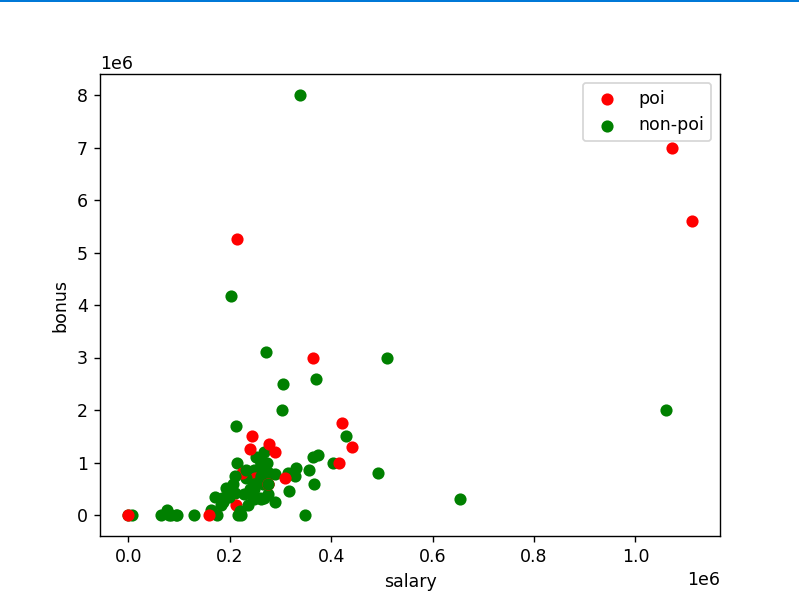


2. **Total Stock Value vs Exercised Stock Options**

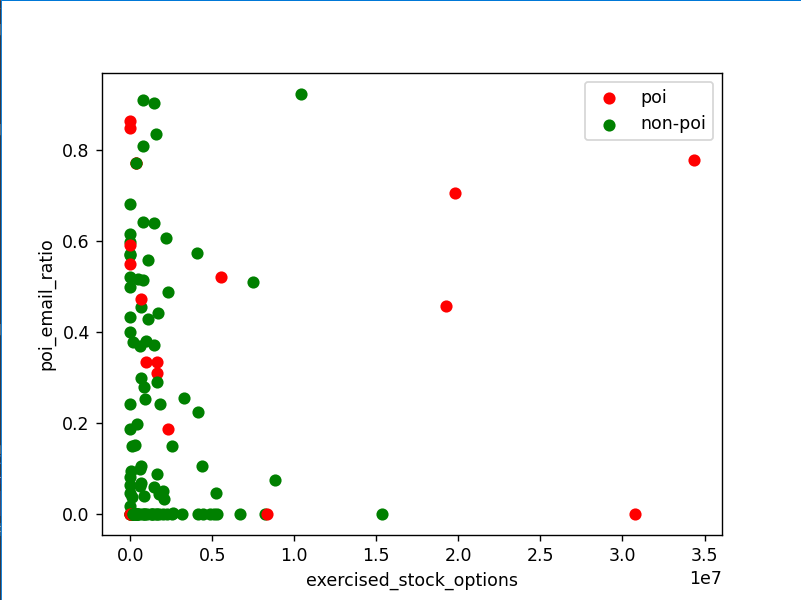


**conclusion :基本上线性相关**

**3. Salary vs Bonus**



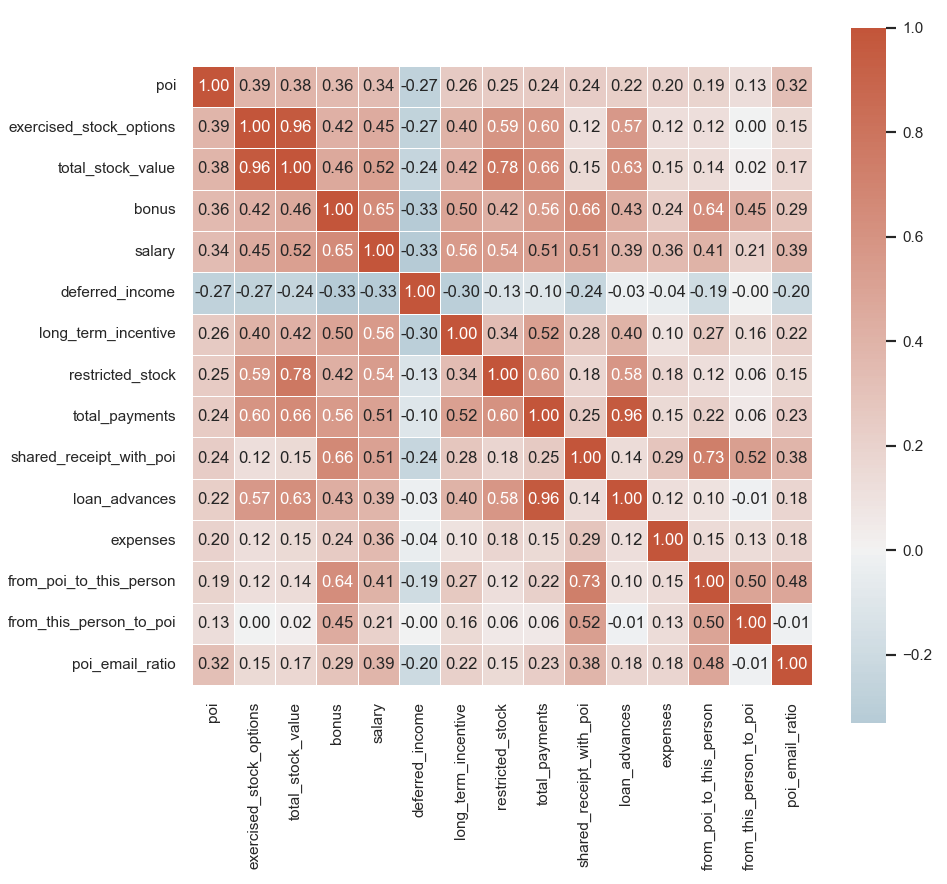
**4.** **exercised\_stock\_options VS poi\_email\_ratio**



这个的意思是 这两个数据无太大关系

无论poi\_email\_ratio大小，与**exercised\_stock\_options的相关性太低**

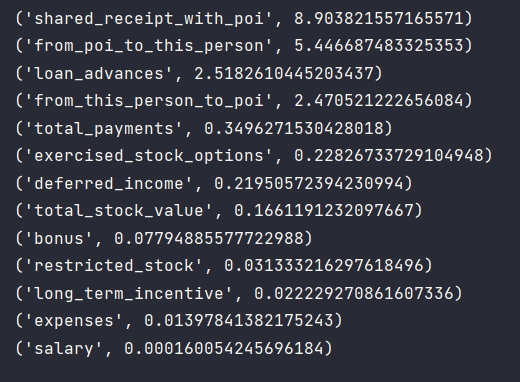
**5.所有特征项 相关系数矩阵图（heatmap）**



## 3.choose top K

使用 Scikit-Learn 的 SelectKBest 算法，简单来说就是算出特征重要性，选择前K个最重要的特征。K的话要选，这个选的依据 report里面写

可以跑下图，但是基本没有参考价值



## 4. Final Algorithm

I decided to try **just 2 features 后面有跑8个feature的图**

**选择这两个feature的理由，修改第168行代码 即可**

**例如**

DT\_features\_list\_email = ['poi', 'exercised\_stock\_options', 'deferred\_income']

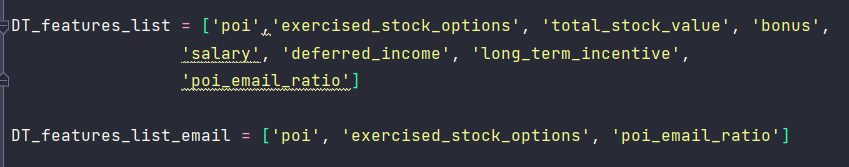
**得出 DecisionTreeClassifier(min\_samples\_split=8)**

**Accuracy: 0.79908 Precision: 0.33370 Recall: 0.30700 F1: 0.31979**

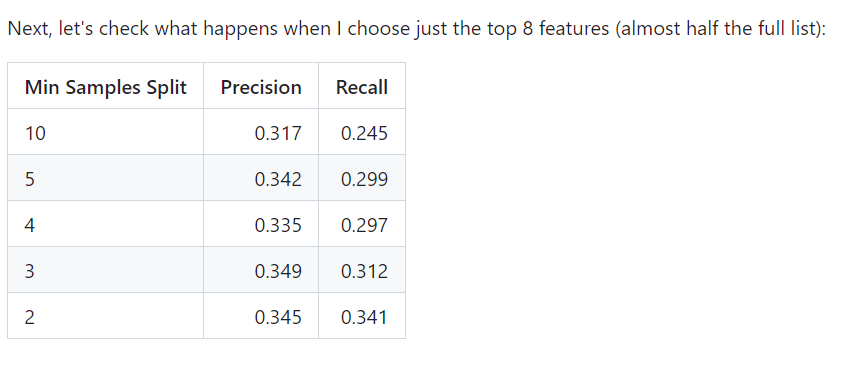
**F1 , precision明显低于最优两个feature**

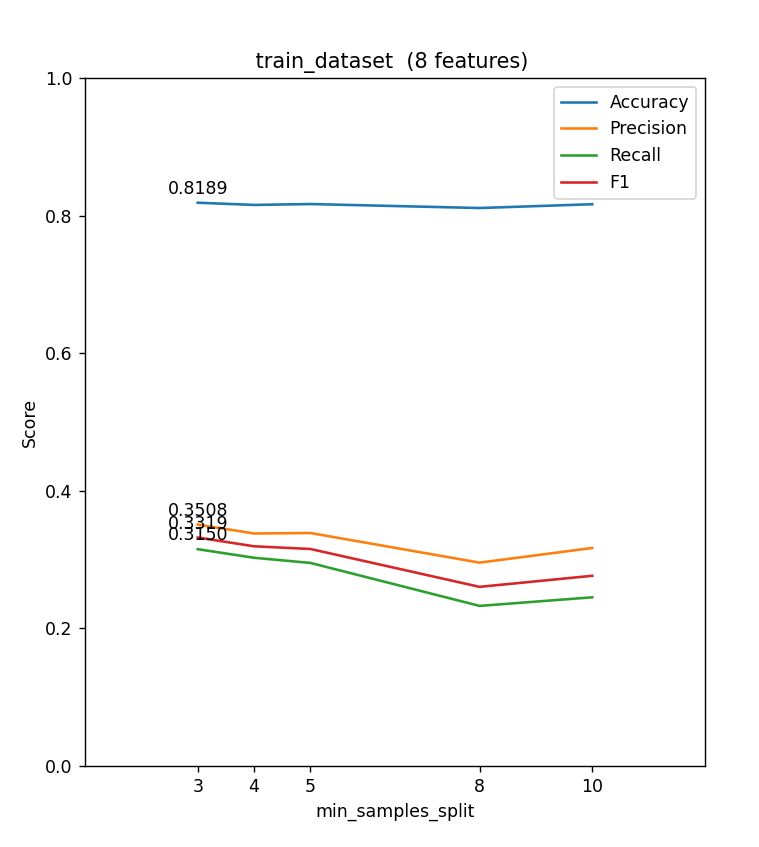
DT\_features\_list\_email = ['poi', 'exercised\_stock\_options', 'deferred\_income']

the best financial feature, 'exercised\_stock\_options', and the best email feature, 'poi\_email\_ratio'. a feature list of ['poi', 'exercised\_stock\_options', 'poi\_email\_ratio'].



原因如下图





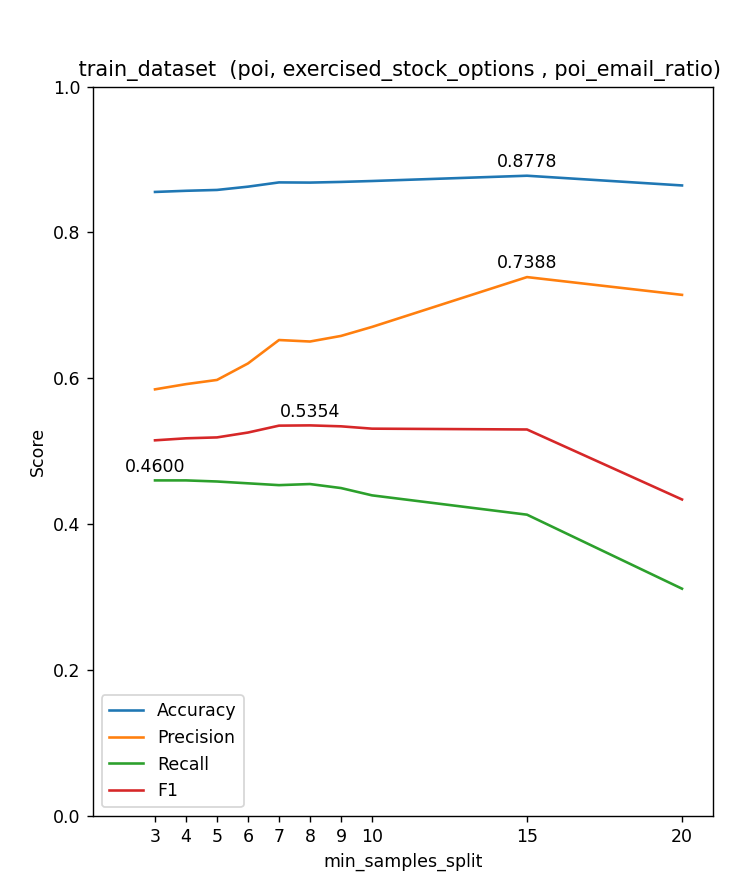
方法：修改 173，193 代码,可以跑8个features的output

DT\_features\_list\_email 🡪 DT\_features\_list

so just use a feature list of ['poi', 'exercised\_stock\_options', 'poi\_email\_ratio'].

The algorithm I've chosen is a **Decision Tree Classifier** with min\_samples\_split = 8

These are the results changing min\_samples\_split:



so min\_samples\_split = 8

选择min\_samples\_split:的原因是 F1的score 最高

综合考虑了精确率和召回率两个指标，因此可以更全面地评估分类模型的性能

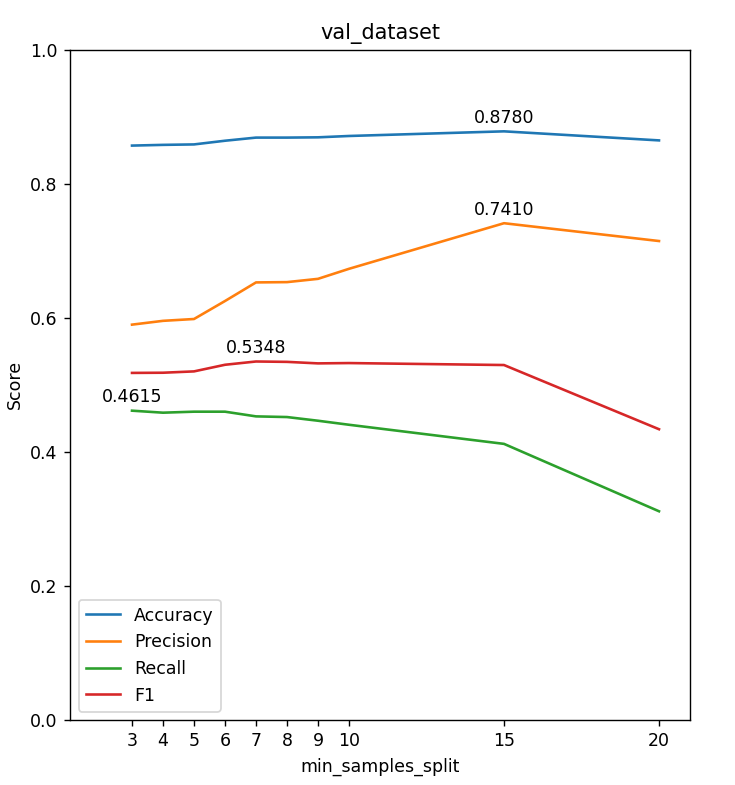
F1 = 2 \* (Precision \* Recall) / (Precision + Recall)

## 5. Validation and Evaluation

用StratifiedShuffleSplit做验证集

**必须在修改完poi.py第107行min\_samples\_split的特定值之后运行poi.py，**

**再运行tester.py即可得出数据输出**



## 6. conclusion

构建的预测模型

根据他们的财务和电子邮件数据出识别可能实施欺诈的安然员工。

