Dataset Summary

Q. total number of data points.

A. Original data has 146 people and 15 features. There should be 146 points if using the raw data

Q. allocation across classes (POI/non-POI)

A. POI : 18, non-POI : 128

Q. number of features used

A. 15 features in data set.
['poi', 'salary', 'deferral_payments', 'total_payments','loan_advances',
'bonus','restricted_stock_deferred','deferred_income', 'total_stock_value', 'expenses','exercised_stock_options',
'other','long_term_incentive',
'restricted_stock', 'director_fees']

Q. Are there features with many missing values? etc.

A. 'restricted_stock_deferred', 'director_fees', 'loan_advances' have almost zero values similar or more than POI values, which are non zero

Feature	non-zero
total_stock_value	19
total_payments	20
restricted_stock	35
exercised_stock_options	43
salary	50
expenses	50
other	52
bonus	63
long_term_incentive	79
deferred_income	96
deferral_payments	106
poi	127
restricted_stock_deferred	127
director_fees	128
loan_advances	141

Q.
Student response identifies outlier(s) in the financial data, and explains how they are removed or otherwise handled. Outliers are removed or retained as appropriate.

A.

'''

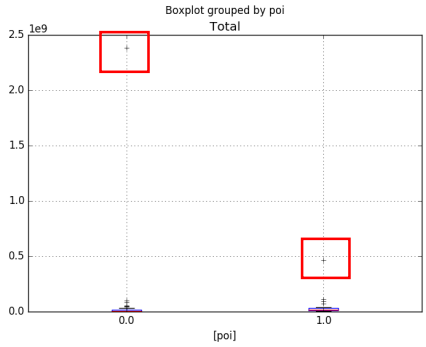
LIST	number of zero in columns
WODRASKA JOHN	15
POWERS WILLIAM	16
THE TRAVEL AGENCY IN THE PARK	15
BROWN MICHAEL	15

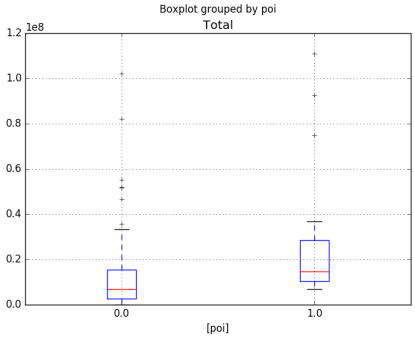
GRAMM WENDY L 15

They have too many zero values inside as many as "THE TRAVEL AGENCY IN THE PARK" which seems like reasonable to remove.

Checked their POI value is zero.
WODRASKA JOHN is POI 0
POWERS WILLIAM is POI 0
THE TRAVEL AGENCY IN THE PARK is POI 0
BROWN MICHAEL is POI 0
GRAMM WENDY L is POI 0
'''

Left plot shows original data box plot which is difficult recognize the spread values.
On the other hand, the processed plot which removed some of high values could be recognized easily how the points are spread.





Q1.
Summarize for us the goal of this project and how machine learning is useful in trying to accomplish it.
As part of your answer, give some background on the dataset and how it can be used to answer the project question. Were there any outliers in the data when you got it, and how did you handle those?
[relevant rubric items: data exploration, outlier investigation]

A1.
This project is to find people who committed a crime with the dataset. How can we find the criminal by getting through our data.
There we use a machine learning, which could figure out some similarities between the criminals along with their personal data.
when we analyzed the criminal information, we could apply those characteristics to the people whether they have in common in some area.
That's how the machine learning works.
While working on the data, there should be outliers that doesn't affect the result finding criminal as well as only mess
up the data consistency. In this data set, there was data that have no values inside and also total values included in the data set.

Q2.
What features did you end up using in your POI identifier, and what selection process did you use to pick them? Did you have to do any scaling? Why or why not? As part of the assignment, you should attempt to engineer your own feature that does not come ready-made in the dataset -- explain what feature you tried to make, and the rationale behind it. (You do not necessarily have to use it in the final analysis, only engineer and test it.) In your feature selection step, if you used an algorithm like a decision tree, please also give the feature importances of the features that you use, and if you used an automated feature selection function like SelectKBest, please report the feature scores and reasons for your choice of parameter values. [relevant rubric items: "create new features", "properly scale features", "intelligently select feature"]

A2.
Actually, this enron dataset already has its own POI even though they are a few. So, here I just needed to pick the right features for the classification and I used selectionKBest mostly, but some algorithm used PCA as well. The first Naïve Bayes didn't need the scale function it was not affected by the scaled value and the result presented the best between three algorithms. Support vector machine, It should be affected by the scale. First time, it didn't work well but when I changed the scale from MinMaxScaler to StandardScaler, it worked fine even though the recall value couldn't make over 0.3. DecisionTree was not affected that much of the scaled dataset.

I created some features for the better result such as

a.
```
data_pd["TotalStock_TotalPay"] = data_pd["total_stock_value"] +
data_pd["total_payments"]
#It had "Total" in common. They might have a relationship if total stock value is
high then total payments might be high as well or not.
```

b.
```
data_pd["Salary_Bonus"] = data_pd["salary"] + data_pd["bonus"]
# It is the genuine value of the employee earned in a year including all the money
someone received from the company. We normally think yearly income as salary and
bonus together.
```

c.
```
data_pd["TotalStock_RestStock_ExercStock"] = data_pd["total_stock_value"] +
data_pd["restricted_stock"] + data_pd["exercised_stock_options"]
# Added all the stock related values
```

SelectKBest accumulated scores

	Accumulated_Score
exercised_stock_options	1.068181e+07
total_stock_value	1.039250e+07
TotalStock_RestStock_ExercStock	1.030532e+07
Salary_Bonus	9.650207e+06
bonus	8.893023e+06
salary	7.707642e+06
TotalStock_TotalPay	7.352396e+06
deferred_income	4.944756e+06
long_term_incentive	4.165918e+06
restricted_stock	3.908955e+06
total_payments	3.821471e+06
loan_advances	3.177775e+06
expenses	2.575345e+06
other	1.815894e+06
director_fees	8.900266e+05
deferral_payments	1.151085e+05
restricted_stock_deferred	3.080609e+04

First choice

Second choice

It seems like if I could choose the data useful, I will choose first seven data first. They look like much higher than the rest underneath it. Secondly, I will decide to use until "other" which seem like the last choice I could have to get a good result I can think of.

Q3.
What algorithm did you end up using? What other one(s) did you try? How did model performance differ between algorithms?
[relevant rubric item: "pick an algorithm"]

A3.
the best answer for the recall and precision values was in the Naïve Bayes classification. I tried to use other classification like Support Vector Machine and DecisionTreeClassifier. DecisionTreeClassifier could make over 0.3 but it is almost close to 0.3. if the value randomly chosen bad, then it might not achieve the target value 0.3.
I used similar processes for all the classification. SelectKBest with GridsearchCV, sometimes PCA together, but I could only get a good result with GaussianNB, DecisionTree which were above 0.3

Q4.
What does it mean to tune the parameters of an algorithm, and what can happen if you don't do this well? How did you tune the parameters of your particular algorithm? (Some algorithms do not have parameters that you need to tune -- if this is the case for the one you picked, identify and briefly explain how you would have done it for the model that was not your final choice or a different model that does utilize parameter tuning, e.g. a decision tree classifier).
[relevant rubric item: "tune the algorithm"]

A4.
Tuning the parameter could be critical when fitting the classifier. Here I tried three classifier. Firstly, I tried to use GaussianNB and needed to tune how many features could result in the best precision and recall value. SelectKBest and PCA were chosen to find the number of features. Also, SupportVectorMachine was tuned with the parameters, kernel, C, and gamma. C and gamma could decide kind of sensitivity of the classification. Kernel could choose the way to divide the data. Parameters for decisionTreeclassifier were max_depth and min_samples_split, which could be similar to C and gamma in SVM.

Q5.
What is validation, and what's a classic mistake you can make if you do it wrong? How did you validate your analysis?
[relevant rubric item: "validation strategy"]

A5.
As I understand, cross validation is to check how the classification could work for others. But there is a case that is hard to get collecting data. So, using the original data, divided it by two different groups. One is for training group which could make the data fit to some classifications and the others are used for a test whether the classification which was fit to training group could be fit in the test group as well.
So, if they are fit in test group, it could work well as other data.

Here, I used stratifiedshufflesplit because there are too few of POI. If I use other cross validation, it might be hard to detect the POI and might just wish for a luck to get a good POI chosen. However, with stratifiedshufflesplit the training data could be randomly selected as I set the fold values and could find the reasonable mix of training set and test set.

Q6.
Give at least 2 evaluation metrics and your average performance for each of them. Explain an interpretation of your metrics that says something human-understandable about your algorithm's performance.
[relevant rubric item: "usage of evaluation metrics"]

A6.

$$\text{Precision} = \frac{tp}{tp + fp}$$

$$\text{Recall} = \frac{tp}{tp + fn}$$

$$\text{Accuracy} = \frac{tp + tn}{tp + tn + fp + fn}$$

TP : The number of True Positives. People who did a fraud and correctly predicted fraud.
TN : The number of True Negatives. People who didn't a fraud and correctly predicted not fraud.

FP : The number of False Positives. People who didn't a fraud and incorrectly predicted fraud.
FN : The number of False Negatives. People who did a fraud and incorrectly predicted not fraud.

DecisionTreeClassifier
Accuracy: 0.80664 Precision: 0.32457 Recall: 0.32700 F1: 0.32578 F2: 0.32651
Total predictions: 14000 True positives: 654 False positives: 1361 False negatives: 1346
True negatives: 10639

: In the decision tree classification,
Accuracy is the probability of correctly predict the fraud and not fraud, which is (654+10639) / 14000 = 0.80.
Precision is the probability of correctly predict the fraud between the people predicted as the fraud whether correctly and incorrectly, which is 654 / (654 + 1361) = 0.32.
Recall is the probability of correctly predict the fraud between the people who made a fraud, which is 654 / (654 + 1346) = 0.327.

GaussianNB

Pipeline(steps=[('select', SelectKBest(k=13, score_func=<function f_classif at 0x1185c0320>)), ('clf', GaussianNB())])

Accuracy: 0.83450 Precision: 0.42332 Recall: 0.43750 F1: 0.43029 F2: 0.43459
Total predictions: 14000 True positives: 875 False positives: 1192 False negatives: 1125
True negatives: 10808

: In case of the GaussianNB classification,

Accuracy is the probability of True, which is (875+10808) / 14000 = 0.83.

Precision is the probability of True&Positive in Positive, which is 875 / (875 + 1192) = 0.42.

Recall is the probability of True&Positive in True&Positive and False&negative, which is 875 / (875 + 1125) = 0.43.