Project 5 Machine Learning

Data Exploration

a. Total number of data points

There are 146 data points in total.

b. Allocation across classes

In the 146 data point, there are 19 pois, 127 non-pois.

c. Number of features

There are 14 financial features, 7 email features.

d. Are there features with many missing values?

“Loan Advance” has almost all of the value missing. Also director fees and restricted stock deferred have a lot of value missing.

Outlier investigation

e. As discussed in the lesson, “TOTAL” is an outlier so I removed it.

f. I noticed that “LOCKHART EUGENE E” has all the value missing. So I removed it as well.

g. “Deferral Payment” should be positive. But on the deferral payments plot, there is one point that is negative. So I wrote the code to print out the deferral payment. I found that for “BELFER ROBERT”, the deferral payment value is negative value -102500. So I check the enron61702insiderpay.pdf and found that it should be NaN for “BELFER ROBERT” in deferral payment. -102500 is his deferred income. So I put this value back to its right place.

h. “restricted stock” should be positive but has one value negative on the plot. It is “BHATNAGAR SANJAY”: -2604490. I checked the document, it should be 2604490 for this field. So I corrected this number.

i. “restricted stock deferred” has all value negative but one positive. This is “BELFER ROBERT”: 44093. The correct value is -44093. So I corrected the number into negative.

j.

Then I noticed some points that seem abnormal. “LAVORATO, JOHN J” has low salary, but very high bonus!!! Even higher than Kenneth lay. “FREVERT, MARK A” has very high salary and deferred income, almost equal to Kenneth. “PAI LOU L” has highest restricted stock, high total stock value. So I googled these persons. I found out that “FREVERT, MARK A” was the chairman and chief executive officer of Enron Europe and north America. So it may be reasonable that he has such high salary. For “LAVORATO, JOHN J”, an article((http://www.nytimes.com/2002/06/18/business/18ENRO.html) said “In Enron's energy-trading unit, for example, John J. Lavorato, a top executive, and John D. Arnold, a gas trader, each received cash bonuses of $8 million to keep them from leaving Enron last fall.” So this explain why he got very high bonus. “PAI LOU L” was the CEO of Enron Energy Services. So it was reasonable he has high total stock value.

Create new features and feature scaling

- New feature
  I checked the plots in this website the coach providec: http://bl.ocks.org/dmenin/raw/d12a22521ad32cacc906/ . I don't think "to_messages", "from_messages" help identify the pois a lot. So I transform it into fraction_ from_to_cc_poi. This new feature equals to the sum of from, to and shared receipt with pois divided by total number of messages.

  I performed SelectKBest to test the feature score. The feature importance score is 8.57, rank 9 in the total 15 features. In the Gridsearch process, the best k=10, the new feature is used in the final model.
- Feature scaling
  I use MaxMinScaler for svm because svm would be affected by feature sclaling. Also, although decision tree and naïve bayes would not be affected, but I deployed pca, which is a process that takes place in Euclidean space so it would require feature scaling.

Intelligently select features

From the Univariate plots(http://bl.ocks.org/dmenin/raw/d12a22521ad32cacc906/), I found most of the features useful, but still some features may not be good:

- "deferral payments" has most of the non-missing value for non-POIS and only a few for POIS.
- "director fees" has all the value for non POIS and all missing values for POIS.
- "loan advance" has only three values and all of the rest are missing.
- "restricted stock deferred" has all non-missing value for non POIS and all missing value for POIS.

To be conservative, I will keep deferral payments and loan advance. Meanwhile I will not use "director fees" and "restricted stock deferred", because I think it may be deceptive.

Therefore, I will select [salary], [deferral payments], [total payments], [loan advance], [bonus], [deferred_income], [total stock value], [expense], [exercised stock option], [other], [long term incentive], [restricted stock], [fraction from to cc poi].

I use SelectKBest to test the importance score of each feature to justify my choice: The features importance scores:

	feature	Feature score
1	**Bonus**	**24.84**
2	**Long term incentive**	**20.58**
3	**Salary**	**20.21**
4	**Total stock value**	**19.23**
5	**Exercised stock options**	**18.97**
6	**Expenses**	**15.68**

7	**Deferred income**	**10.64**
8	**Total payments**	**8.95**
9	**Fraction from to cc pois**	**8.57**
10	**Loan advances**	**7.83**
11	**Other**	**7.80**
12	Restricted stock	6.02
13	Deferral_payments	0.61
14	Director fees	1.48
15	Restricted stock deferred	0.09

This score proves my judge by the plots. Deferral payment, director fees and restricted stock deferred have very low feature importance score.

In the later process, Gridsearch select the best k=10 for SelectKBest . The ten features selected are those have higher importance score.

Therefore, the 10 features selected are:
['salary', 'total_payments', 'bonus', 'expenses', 'deferred_income', 'total_stock_value', 'long_term_incentive', 'loan_advances', 'exercised_stock_options', 'fraction_from_to_cc_pois']

Pick and tune an algorithm

- Pick an algorithm

I use three algorithms: naïve bayes, svm and decision tree. For each of these algorithm, I got the performance as followed:
Naïve bayes:
Accuracy: 0.82, precision: 0.14, recall: 0.09
Svm:
Accuracy: 0.86, precision: 0.14, recall: 0.05
Decision tree:
Accuracy: 0.75, precision: 0.14, recall: 0.20

Accuracy measures the fraction of correct predicted points on all the test points. Precision measures the fraction of actually positive points when the prediction is positive. When the precision is high, it means when we predict a person is poi, there is a very high probability that this person is actually a poi. Recall measures the fraction of predictive positive when it is actually positive. When recall score is high, it means when a person is poi, we have a very large probability to label this person as poi.

Decision tree got the highest precision and recall score, so I will pick decision tree as my algorithm.

- Tune parameter

Parameter tuning is to select the optimized parameter that improve the performance of a learning algorithm. It is very important because tuning parameter could ensure the model does not overfit its data.

I used GridSearchCV to tune the algorithm. I got the pca__n_components = 2, k = 10 for SelectKBest, min_sample_split = 10.

The 10 features selected are:
['salary', 'total_payments', 'bonus', 'expenses', 'deferred_income', 'total_stock_value', 'long_term_incentive', 'loan_advances', 'exercised_stock_options', 'fraction_from_to_cc_pois']

The "fraction_from_to_cc_pois", the new feature we created, is judged important by the SelectKBest, with the important score 8.57.

The model I used eventually is:

```
features_list = ['poi', 'salary', 'total_payments', 'bonus', 'expenses',
                 'deferred_income','total_stock_value','long_term_incentive',
'loan_advances','exercised_stock_options', 'fraction_from_to_cc_pois']

estimators = [('pca', PCA(n_components=2)),
('tree',tree.DecisionTreeClassifier(min_samples_split=10))]
clf = Pipeline(estimators)
```

I got precision = 0.35, recall = 0.31 in the tester.py.

Validation

Validation is to split the data set into train and testing data, using the test data to validate the clf. Testing gives estimate of performance on an independent dataset and serves as check on overfitting. In this project, I have split the data set into 0.7 training data and 0.3 testing data.