Enron. Identifying fraud.

This project takes a closer look on the Enron fraud case revealed towards the end of 2001.

Background.

In its best years Enron was defined by many as innovative, ahead of its time, daring and revolutionary.

The company was formed in 1985 following a merger between Houston Natural Gas Co. and Omaha-based InterNorth Inc. As the chief executive officer (CEO) of Houston Natural Gas, Kenneth Lay became Enron's CEO and chairman, and quickly rebranded Enron into an energy trader and supplier. Enron took great advantage of the deregulation of the energy markets which allowed companies to place bets on future prices.

What allowed Enron to flourish was the dot-com bubble at the end of the 90s when the bubble was in full swing, and the Nasdaq hit 5,000. Revolutionary internet stocks were being overvalued and regulators simply accepted spiking share prices as the new normal.

[1]

Exploration of the dataset

The studied dataset has a total of 146 entries.

Several observations are marked as "POI" which means that they were persons of interest during the legal investigation of the Enron fraud.

Some researches claim that the extent of fraud and speculations at Enron was driven by the management always demanding from new strategies, innovation and being the best on the market without scrutinizing the methods and means for achieving that.

---

[1] Read more: Enron Scandal:
The Fall of a Wall Street Darling | Investopedia http://www.investopedia.com/updates/enron-scandal-summary/#ixzz4M3BnGYwZ
The Rise and Fall of Enron | Journal of Accountancy
http://www.journalofaccountancy.com/issues/2002/apr/theriseandfallofe nron.html

A list of the 18 POI in this dataset is presented below.

Person of interest in the Enron dataset
POI: HANNON KEVIN P
POI: COLWELL WESLEY
POI: RIEKER PAULA H
POI: KOPPER MICHAEL J
POI: SHELBY REX
POI: DELAINEY DAVID W
POI: LAY KENNETH L
POI: BOWEN JR RAYMOND M
POI: BELDEN TIMOTHY N
POI: FASTOW ANDREW S
POI: CALGER CHRISTOPHER F
POI: RICE KENNETH D
POI: SKILLING JEFFREY K
POI: YEAGER F SCOTT
POI: HIRKO JOSEPH
POI: KOENIG MARK E
POI: CAUSEY RICHARD A
POI: GLISAN JR BEN F

For each person on the list there are up to 21 entered features partially derived from and specified by the Summary Schedule of all Debtors Combined.´

Example:

METTS MARK
{'to_messages': 807, 'deferral_payments': 'NaN', 'expenses': 94299, 'poi': False, 'deferred_income': 'NaN', 'email_address': 'mark.metts@enron.com', 'long_term_incentive': 0.0, 'perks_ratio': 0.941684594408002, 'restricted_stock_deferred': 'NaN', 'poi_ratio_messages': 0.8863636363636364, 'shared_receipt_with_poi': 702, 'loan_advances': 0.0, 'from_messages': 29, 'other': 1740, 'director_fees': 0.0, 'bonus': 600000, 'total_stock_value': 585062, 'from_poi_to_this_person': 38, 'from_this_person_to_poi': 1, 'restricted_stock': 585062, 'salary': 365788, 'total_payments': 1061827, 'employee_perks': 1550850.0, 'exercised_stock_options': 0.0}

There is missing information about most of the people on the list.

The only feature without missing information is poi while the feature with missing information about most of the observations is loan advances.

Number of missing values per feature:	
loan_advances	142
director_fees	129
restricted_stock_deferred	128
deferral_payments	107
deferred_income	97
long_term_incentive	80
bonus	64
from_this_person_to_poi	60
from_poi_to_this_person	60
from_messages	60
shared_receipt_with_poi	60
to_messages	60
other	53
expenses	51
salary	51
exercised_stock_options	44
restricted_stock	36
email_address	35
total_payments	21
total_stock_value	20
poi	0

A logical explanation about the larger number of missing observations for loan advances, restricted stock and deferral payments could be that they are more typical for executives and managers and non-employee directors.

- Deferral payment is an executive perk used from executives to avoid/delay paying taxes.

- Director fees is paid to non-employee directors.

- As stated in the Report of investigation of Enron corporation and related entities regarding federal tax and compensation issues and policy recommendations, deferral of restricted stock was available only for executives and non-employee directors who could choose to not receive a share upon maturity of an option but instead Enron would credit the value to the participant's Phantom Stock Account that will be payable upon death or retirement due to disability. Credits for the dividends were supposed to be accrued in a separate account.[2]

[2] Sources:

http://www.bloomberg.com/news/articles/2002-03-03/the-danger-of-deferred-compensation

https://books.google.lt/books?id=yhSA2u91BFgC&pg=PA610&lpg=PA610&dq=restricted+stock+deferred+enron&source=bl&ots=Nb08uMf0vm&sig=OXZPsqrYA0ccumN2zHAZ48mwk3s&hl=en&sa=X&redir_esc=y#v=onepage&q=restricted%20stock%20deferred%20enron&f=false

Presented below is a list of the top 5 entries with least amount of information.

LOCKHART EUGENE E has only been indicated as no person of interest and all the rest of the information is missing. I see no real reason for keeping this observation in the dataset.

Top 5 of the people with least amount of information:	
LOCKHART EUGENE E	20
GRAMM WENDY L	18
WROBEL BRUCE	18
WHALEY DAVID A	18
THE TRAVEL AGENCY IN THE PARK	18

3 of the main investigated people during the Enron fraud scandal were:

- Kenneth Lay – former CEO and Chairman of Enron.
- Andrew Fastow - former CFO of Enron indicted on 78 counts of securities fraud, money laundering, wire and mail fraud, as well as conspiracy to inflate Enron's profit.
- Jeffrey Skilling - former CEO of Enron.[3]

The total payments that each of them has reportedly received is:

Total payments to Kenneth Lay : 103559793 USD

Total payments to Andrew Fastow : 2424083 USD

Total payments to Jeffrey Skilling : 8682716 USD

Outliers

Next I checked for any possible outliers.

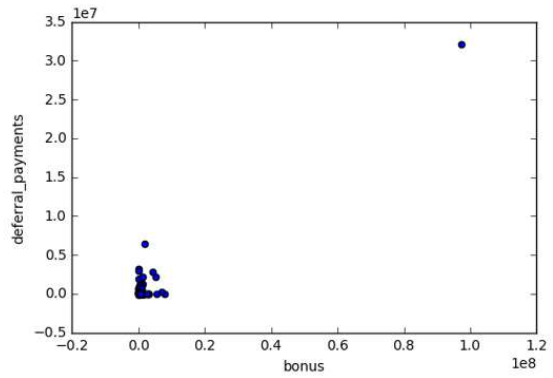
By plotting two of the features of interest: bonus and deferral payments I immediately saw that there was a big outlier disturbing the results for the entire dataset. After closer look I found out that this observation is the TOTAL which meant that it could be safely removed.

Another outlier I noticed after a closer look at the top 5 of the people with least amount of information was THE TRAVEL AGENCY IN THE PARK. This obviously is not an employee so I decided to remove it from the dataset.

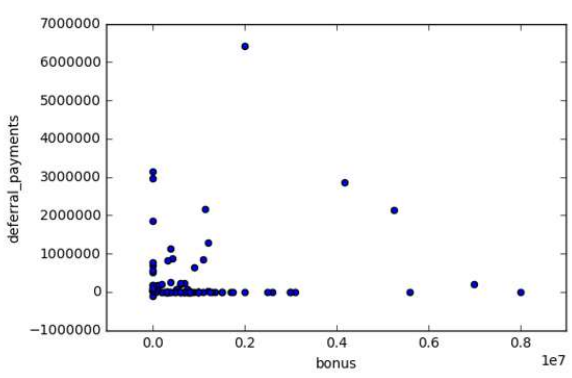
I also decided to remove LOCKHART EUGENE E since the only information about this person was that he is not a person of interest.

[3] Source: http://www.lawyershop.com/practice-areas/criminal-law/white-collar-crimes/securities-fraud/lawsuits/enron

Before removing TOTAL:



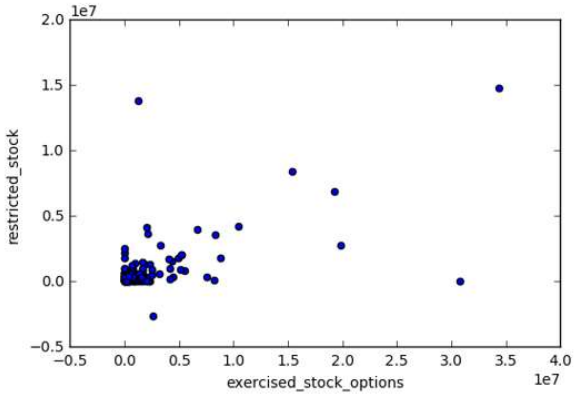
After removing TOTAL:



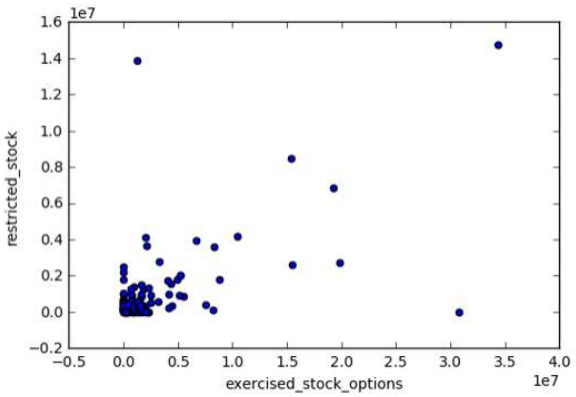
After that I plotted also exercised stock options and restricted stock and I noticed that the observations for BHATNAGAR SANJAY and BELFER ROBERT seemed to not match the information from the Summary Schedule of all debtors combined.

I made some manual adjustments for those two entries so that they will match the provided Summary Schedule.

Before correcting the observations:



After correcting the observations:



New features

Next I proceed with the creation of new features.

1. I created a ratio of POI messages to total amount of messages each person has handled employees
2. I created an aggregate of all employee perks given to the most successful employees and especially executives. According to most of the researching articles they together with the corporate culture were the main motives for fraud.
3. Perks ratio which is the employee perks compared to total earnings.

Feature selection

Before proceeding with the testing and tuning of my classifiers of choice, I needed to choose the best features out of an existing list. In order to do that I first had to split the data in one part that was going to be used for training and another for testing.

I mimicked the code from tester.py where the data was split with the help of StratifiedShuffleSplit.

StratifiedShuffleSplit splits the data into multitude of training and testing (validating) parts where the number of folds equals 'n_iter'. For each of the folds StratifiedShuffleSplit chooses different points within the dataset.

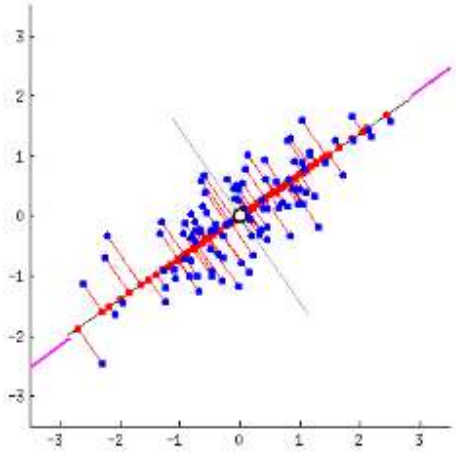
The created training and testing data I used further in the selection process of the best features.

The used classifier is actually a pipeline where I used scaling with MinMaxScaler(), choosing the features with the top scores of the performed ANOVA analyses with the help of SelectKBest(). In order to reduce dimensionality I used PCA() which was followed by the classifier SVM().

For finding the optimal combination of parameters for all the components of the pipeline I used GridSearchCV where each combination was evaluated according to its F1 score.

F1 score is the weighted average of the precision (true positives/(true positives + false positives) or the number of times the classifier has guessed correctly that somebody is a person of interest divided by all the guesses that somebody is a person of interest) and recall (true positives / (true positives + false negatives) or the number of times the classifier has guess correctly that somebody is a person of interest divided by the total number of persons of interest).

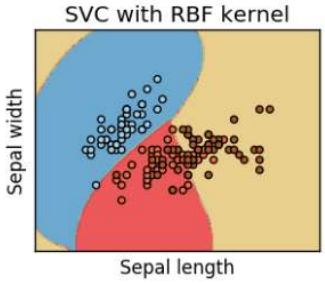
The MinMaxScaler() was used to prepare the data for the principle component analysis. It transformed the data into values ranging from 0 to 1 which was necessary in order to save the information from all the observations with 0 and to minimize the risk of PCA() giving biased results. PCA computed n number of new components (lines), based on the original data, that successfully fit all the original data points so that the distance between each of those points and the component(the line) was at its minimum. It used a similar approach as the OLS.



Example:

Source: http://stats.stackexchange.com/questions/2691/making-sense-of-principal-component-analysis-eigenvectors-eigenvalues

The selected svm.SVC() classifer was with rbf kernel which means that the separation of the data looked something similar to this:



The Support vector machines focus only on the points that are the most difficult to tell apart, whereas other classifiers pay attention to all of the points.

The used C parameter controlled the tradeoff between smoother decision boundary and classifying the points correctly.

Source: http://stats.stackexchange.com/questions/23391/how-does-a-support-vector-machine-svm-work

The validation of my model was done by GridSearchCV. The chosen features were:

['salary', 'bonus', 'total_stock_value', 'exercised_stock_options', 'restricted_stock', 'employee_perks']

The flow of the entire feature selection process was as follows:

stratifiedshufflesplit -->
pipeline --> processor , SelectKBest ,SVC
--> Train
--> Validate
--> Repeat

The beauty of the cross validation with GridSearchCV is that it evaluates many different combinations of parameters with only couple of extra rows with code which essentially means that it does the parameter tuning for you.

What is parameter tuning

Each classifier (apart from simple ones like Naive Bayes), has parameters that you can select. Selecting the parameters that lead to the model with the most optimal results is parameter tuning.

I did my parameter tuning using GridSearchCV, where I specified in a 'grid' a number of different parameters which GridSearchCV fitted and returned the optimal model according to the chosen criteria F1 score.

Model selection and tuning

In order to control the results for the best features I hardcoded the list of new features and split it the same way I did with my initial list of intuitively selected features + created ones.

I used the GausianNB() classifier as a method of controlling the goodness of my results. As mentioned earlier this is one of the simplest classifiers where there is no need of parameter tuning.

The obtained results were as follow:

Classifier	**Accuracy**	**Precision**	**Recall**	**F1**	**F2**
GaussianNB	0.84421	0.43522	0.30400	0.35796	0.32351

Model evaluation/cross validation

The model was evaluated with the help of tester.py where just as in my project the information was first imported, then split in n number of folds with the help of StratifiedShuffleSplit. By definition it is a variation of ShuffleSplit, which returns stratified splits, i.e which creates splits by preserving the same percentage for each target class as in the complete set. This means that the usual split between training, validation and testing data is replaced by taking small pseudo-random parts of each data set for validation and testing. In this case a 1000 number of folds were created with pseudo-randomness in the choice of different data points for each fold equal to 42.

After that the created folds were split into testing and training features and labels and fitted with the created classifier.

The goodness of fit of the classifier was then evaluated by sorting the results in 4 different categories:

1. True_negative - points (people) that the classifier labeled as no person of interest and they in fact were no person of interest
2. False_negative – points (people) that the classifier labeled as no person of interest when they in fact were a person of interest
3. Fals_positive – points (people) that the classifier labeled as a person of interest when they in fact were not such
4. True_positive – when the classifier successfully labeled a person of interest

After sorting each point in such manner the results are further evaluated by calculating the accuracy, precision, recall, f1 and f2.

1. Accuracy – the sum of the true_positive and true_negative labels divided with the sum of all labels returned by the classifier
2. Precision – the true_positive labels divided with all the poisitive labels returned by the classifier or the proportion of correctly labeled persons of interest out of all the points that the classifier counted as a person of interest
3. Recall – the true_positive labels divided with all sum of true_positive and false_negative labels returned by the classifier i.e. the portion of correctly labeled persons of interest out of all the persons of interest that existed in the data.
4. F1 – the weighted average of the precision and recall metrics where the best score is 1 and the worst is 0
5. F2 – is also weighted result of precision and recall giving more weight to the recall metric. Here again the best result is 1 and worst is 0.

This meant that the results I got for my simplest classifier were below average. The main reason is the small number of persons of interest in the dataset and the multitude of missing information. The results would have been even worse if I did not create the 1000 folds of different combinations of data points with the help of StratifiedShuffleSplit.

Tuning my other two classifiers of choice

My other two classifiers of choice were the DecisionTreeClassifier and the LigisticRegression.

I repeated the same process of parameter tuning as with the feature selection.

For the Logistic regression I got:

```
Pipeline(steps=[('minmax', MinMaxScaler(copy=True, feature_range=(0, 1))), ('PCA', PCA(copy=True, n_components=4, whiten=False)), ('clf', LogisticRegression(C=50, class_weight='balanced', dual=False,fit_intercept=True, intercept_scaling=5, max_iter=100,multi_class='ovr', n_jobs=1, penalty='l2', random_state=None,solver='liblinear', tol=1e-05, verbose=0, warm_start=False))])
```

And for the Decision Tree Classifier I got:

```
Pipeline(steps=[('DecisionTreeClassifier', DecisionTreeClassifier(class_weight=None, criterion='entropy', max_depth=None, max_features=None, max_leaf_nodes=20, min_samples_leaf=2,min_samples_split=10, min_weight_fraction_leaf=0.3,presort=False, random_state=20, splitter='best'))])
```

For the sake of speeding up the evaluation process I hardcoded both optimal classifiers.

At the evaluation stages I used again the test_classifier function in tester.py.

For my optimal Logistic Regression model which I obtained using Grid Search I got the following results:

```
Accuracy: 0.74600   Precision: 0.27449   Recall: 0.47350   F1: 0.34752   F2: 0.41354
        Total predictions: 14000          True positives:  947   False positives: 2503   False negatives: 1053

True negatives: 9497
```

Here the values for F1, F2 and especially recall were good enough for the purpose of this project but the precision was rather low which meant that my classifier was not doing well at correctly identifying someone as person of interest.

I experimented further with the parameters of my classifier until I found out that the results were mostly influenced by the number of principle components. The experimentation process is not documented. I got drastically higher results when I left only one principal component. Which meant that all the features were summed up in 1 line. Below I list the obtained results:

```
Pipeline(steps=[('minmax', MinMaxScaler(copy=True, feature_range=(0, 1))), ('PCA', PCA(copy=True, n_components=1, whiten=False)), ('clf', LogisticRegression(C=50, class_weight='balanced', dual=False,fit_intercept=True,
intercept_scaling=5, max_iter=100,multi_class='ovr', n_jobs=1, penalty='l2', random_state=None,solver='liblinear',
tol=1e-05, verbose=0, warm_start=False))])
Accuracy: 0.77507   Precision: 0.32846   Recall: 0.55000   F1: 0.41129   F2: 0.48462
        Total predictions: 14000   True positives: 1100   False positives: 2249   False negatives:  900
        True negatives: 9751
```

I got above average results for my recall which meant that the chosen classifier was successfully identifying persons of interest in more than 50% present of the time. The accuracy of my classifier improved as well.

I did not choose this as my final classifier since I do not fully understand the reason behind this quite substantial change in the results and why the Grid Search has ignored this parameter combination.

For my Decision Tree Classifier I got the following results:

Accuracy: 0.82500	Precision: 0.37054	Recall: 0.32200	F1: 0.34457	F2: 0.33066
Total predictions: 14000		True positives: 644	False positives: 1094	False negatives: 1356
True negatives: 10906				

The high accuracy meant that most of the observations were correctly labeled. This result is greatly influenced by the large number of people correctly identified as no person of interest.

The lower results of precision and recall mean that approximately only 1/3 of the persons of interest were correctly identified.

These results are greatly influenced by the type of unbalanced information that was fed to the classifier. It is rare though that nicely balanced and clean dataset can be obtain in the real world.

The further experimentation with the feature and model selection process I left out of this project.

Disclaimer

My feature and model selection strategies have shifted from more intuitive to more technical since the previous submission. I would continue experimenting with different data sources and searching for the perfect combination of technical knowledge and intuition.