

Intro to Machine Learning Final Project - Table of content

Enron Submission Free-Response Questions

By Raju Nanduri, September 2016

1. Summarize **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"]
2. 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"]
3. 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"]
4. 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"]
5. 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"]
6. 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"]



Some background information

How the Case Was Won

Prosecutors moved methodically up the chain of responsibility, using the word of insiders to tie Lay and Skilling to a scheme to boost Enron's earnings.

SIX KEY FIGURES

1. **Richard A. Causey**, former Enron chief accounting officer, broke from a unified defense team with Kenneth L. Lay and Jeffrey K. Skilling when he pleaded guilty to securities fraud in December.

2. **Andrew S. Fastow**, former finance chief, facing numerous criminal counts, pleaded guilty to two conspiracy charges in exchange for cooperation.

3. **Lea W. Fastow**, former assistant treasurer and Fastow's wife, pleaded guilty to a tax charge, cementing her husband's cooperation.

4. **Ben F. Glisan Jr.**, former treasurer, pleaded guilty to conspiracy in September 2003 and began cooperating with prosecutors in early 2004.

5. **Mark E. Koenig**, former head of investor relations and the first prosecution witness, pleaded guilty in August 2004 to aiding and abetting securities fraud. Jurors singled out his and Glisan's testimony as the most persuasive.

6. **Michael Kopper**, former top Fastow aide, was the first company insider to plead guilty.

Color Key

- Pled guilty and testified against Skilling and Lay
- Pled guilty but did not testify
- Former chiefs convicted

OTHERS WHO PLEADED

7. **Paula H. Rieker**, former No. 2 executive in investor relations and corporate secretary.

8. **David W. Delainey**, former head of Enron's trading and money-losing retail energy units.

9. **Kenneth D. Rice**, former broadband unit CEO.

10. **Kevin P. Hannon**, former chief operating officer for the broadband unit.

11. **Timothy N. Belden**, former top Enron trader.

12. **Timothy Despain**, former assistant treasurer.

13. **Christopher Calger**, former vice president in the trading unit.

14. **John Forney**, former energy trader.

15. **Jeffrey S. Richter**, former trader.

16. **Lawrence Lawyer**, former finance executive.

Enron was a massive failure, partly because of its size, partly because of its complexity, partly because the controls to protect the integrity of capital markets failed, and **especially because of the massive greed and collusion of key participants**. Management failed, auditors failed, analysts failed, creditors/bankers failed, and regulators failed. The intersection of multiple failures sent a signal of structural problems.

Question 1

Summarize 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”]

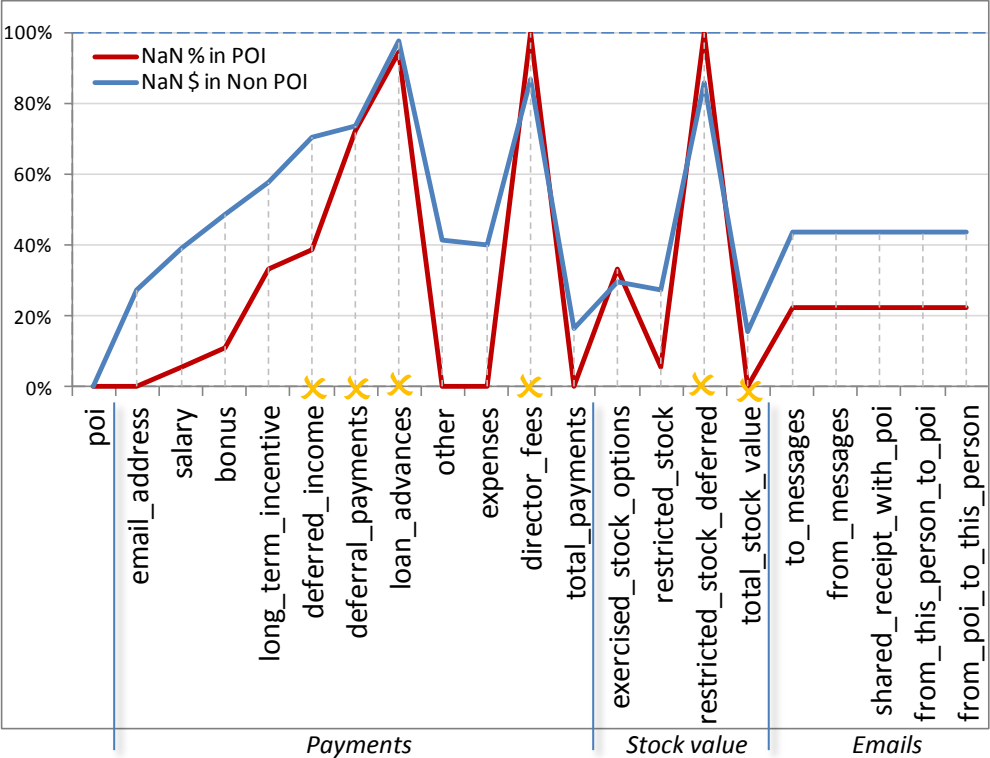
Goal is to **build a person of interest identifier** based on financial and email data from set of Enron employees and select data points (features) and list of known of POIs (labels). **Persons of interest in the fraud case** are those individuals who were indicted, reached a settlement or plea deal with the government, or testified in exchange for prosecution immunity.

Supervised classification learning methods are suitable here given set of features and set of labels.


The data set in the **final_project_dataset.pkl** has 146 records each of which has 21 data points each (features). Nearly 45% of the data is not available i.e. is marked as NaN.

- Given the small size of the data set and for ease of use, I imported the dataset into Excel to explore the data and also identify any obvious outliers. Some of the findings (see below charts as well):
- Features with “richest” data quality (i.e. lowest number of NaN): total payments and total stock value but given it is sum not to be considered
 - Records with “poorest” data quality (i.e. highest number of NaN): Lockart Eugene (20), Whaley David (18), Wrobel Bruce (18), The Travel Agency in the Park (18), Gramm Wendy (18). Of these the “Travel agency in the park” seems odd (human intuition).
 - Reviewing the outliers in the two features with most data, the record labeled as “Total” is clearly an outlier. The other record is Lay Kennth. But again given background information, Ken Lay being the CEO this is not an outlier.
 - Interestingly 60 persons do not have any data re emails including four POIs of which critically Andrew Fastow’s email data is missing

The **number of POI is 18 (around 12% of the 146 records)**. Training the alogrithm with such sparse data set and coupled with poor data quality (nearly 50% of data points are unavailable), will be challenging.



My additional observations:

- I discarded immediately features  with significant missing data resulting in limited discriminating power. Note: possibly can be finetuned for a more granular new composite feature
- In general Non POI have higher NaN other than exercised_stock_options. Tying this to compensation (salary + bonus) would possibly a good feature to label folks who are aggressively “offloading”
- Email data is suprisingly incomplete given this was derived
- Imputation is mostly near to impossible for most data points. E.g. Expenses: Initially I assumed NaN would mean being a NonPOI there are not senior folks. But Dana Gibbs is a counter example.

Question 1 (additional inputs)

I asked myself why identifying persons of interest who were implicated in the Enron scandal was useful.

I believe this is useful as it ties back to one if not the key reason for the sudden mettle-down at Enron: **greed from individual executives** to unabashedly make money while fully in knowledge of the dire straits the firm was at that time.

The critical instrument used for this was stock options. By not having stock options show up in the balance sheet, executives with insider information were able to use it to their benefits at the cost of others who did not have the information. Though not explicitly an outcome for this project, identifying additional persons of interest can help to predict missed POIs (Enron had 20'600 staff with nearly 62% of 401(k) assets in Enron stock).

Below chart shows how things accelerated at Enron.

Side note: what is incomprehensible to me is how such smart people (POIs) believed they can play both sides (see highlighted in red). Also, when did the POIs know of the state of affairs?



- A) **1996 to 2001:** Fortune magazine named Enron "America's Most Innovative Company" for six consecutive years.
- B) **1999 to mid 2001:** A group of 29 Enron executives and directors received \$1.1 billion **by selling 17.3 million shares from 1999 through mid-2001**, according to court filings based on public records.
- C) **17 Apr 2001:** Enron **announces \$536 million in profits for the first quarter**.
- D) **14 Aug 2001:** When **Jeffrey K. Skilling suddenly resigns** as chief executive, citing "personal reasons," Mr. Lay retakes the job. He says, "Absolutely no accounting issue, no trading issue, no reserve issue, no previously unknown problem issues" are behind the departure.
- E) **20 Aug 2001:** Kenneth Lay, the CEO sells 93,000 shares, earns \$2m; urges employees to buy company shares.
- F) **26 Sep 2001:** In an online chat with employees, Mr. Lay says that Enron stock is a good buy and that the company's accounting methods are "legal and totally appropriate."
- G/H/I) Enron reports third quarter loss in Q3 2001 and then files for bankruptcy in Oct 2001

Question 2

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"]

1. After discarding features with significant lack of data like loan advances, director fees etc (see slide 4) I picked from each category type (payments, stock and emails) I zeroed in on set of features based on intuition:
- 'salary', 'bonus', 'other': given POIs are all employees I believe this is a good starting position. I included other given income for POIs typically higher
 - 'long_term_incentive': should be higher for executives to avoid incentivizing gaming for short term gains
 - 'exercised_stock_options': given POIs were those with insider info, this seems like a crucial feature
 - 'shared_receipt_with_poi': if poi shares lot of emails to this person then this person could be also a poi
2. In a second step I reviewed using scatterplots and excel to understand quality, spread and also correlation of the data points (see next page). This helped me in indentifying meaningful composite features and also reject some of the earlier features I had:
- 'salary'+ 'bonus' (total comp) : Combining for better some directional imputation and also cancels out the outliers in each feature.
 - Ratios reviewed:
 - Excercised options to total_comp: to identify POIs in the lower bracket who have high ratio of offloading their options. Note: there was one significant outlier.
 - Exercised options to long term incentives: Typically those with higher long term incentives would not offload their options and if they do despite then suspicion of POIs
 - From poi to this person by shared receipt with POI: relating the shared emails to those being sent from POIs can help to make the classification more precise
3. In the third step, I reviewed my intuitive choices of features by computing the feature importance. I tried total of 5 feature sets of which two I "built" iteratively using cross validation of train test split and stratified shuffle split respectively. I also used SelectKBest with k=4,3 and 2 resp. Reviewing the resulting metrics I finally picked feature set 5: exercised_options and totComp

Note: CV = train_test_split (test size =0.3, random state = 42) and clf = DecisionTree(min samples = 10)

		Measures				Feature importance									
My iterative review using cv=SSS		Accuracy	Precision	Recall	F1	salary	bonus	other	long term incentive	exercised options	shared receipt with POI	totComp	exerOpt to totComp	exerOpt to long Term Incentive	from POI by shared receipt
Iteration	Run 1	0.775	0.000	0.000	0.000	0.000	0.000	0.441	0.091	0.440	0.027				
	Run 2	0.900	0.500	0.250	0.333	0.000	0.000	0.034	0.000	0.480	0.054	0.201	0.000	0.230	0.000
	Run 3	0.872	0.400	0.500	0.444	x	x	x	x	0.297	0.192	0.400	x	0.111	x

Total compentation = Salary+Bonus+Other

Feature set 1

Note: CV = StratifiedShuffleSplit(folds=1000, random state = 42) and clf = DecisionTree(min samples = 10)

		Measures				Feature importance									
My iterative review using cv=SSS		Accuracy	Precision	Recall	F1	salary	bonus	other	long term incentive	exercised options	shared receipt with POI	totComp	exerOpt to totComp	exerOpt to long Term Incentive	from POI by shared receipt
Iteration	Run 1	0.801	0.254	0.201	0.225	0.077	0.278	0.207	0.000	0.339	0.099				
	Run 2	0.819	0.364	0.358	0.361	0.000	0.000	0.137	0.040	0.326	0.046	0.146	0.304	0.000	0.000
	Run 3	0.847	0.503	0.423	0.459	x	x	0.179	x	0.237	x	0.355	0.229	x	x

Total compentation = Salary+Bonus+Other

Feature set 2

Note: CV = StratifiedShuffleSplit(folds=1000, random state = 42) and clf = DecisionTree(min samples = 10)

Select kBest		Measures				Feature importance									
		Accuracy	Precision	Recall	F1	salary	bonus	other	long term incentive	exercised options	shared receipt with POI	totComp	exerOpt to totComp	exerOpt to long Term Incentive	from POI by shared receipt
Iteration	k=4	0.837	0.463	0.364	0.407		0.095	0.064		0.367		0.474			
	k=3	0.834	0.453	0.382	0.414		0.177			0.458		0.366			
	k=2	0.840	0.525	0.426	0.470					0.510		0.490			

Total compentation = Salary+Bonus+Other

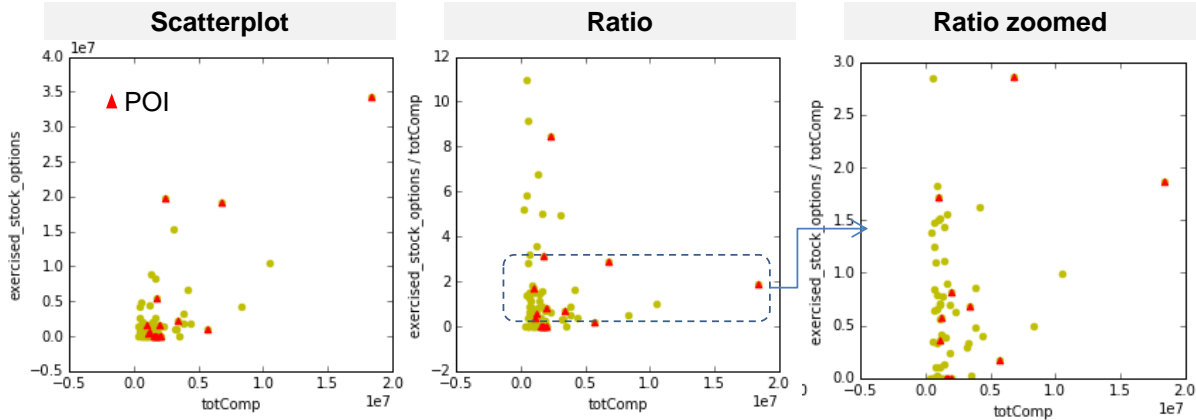
Feature set 3
Feature set 4
Feature set 5

Question 2 (additional inputs)

Key charts highlighted used to validate my intuition on composite features

Total compensation (salary , bonus, other) and exercised stock options

- 1. Exercised stock options would be higher for POIs esp when they are aware of events unfolding against their favor i.e. sell. Comparing this against total compensation highlights those who despite high income are selling.
- 2. Ratio of stock options exercised to compensation on the other hand is to capture those in the lower income bracket who might be aggressively offlading their options. Excl. one significant outlier, there are some patterns emerging.

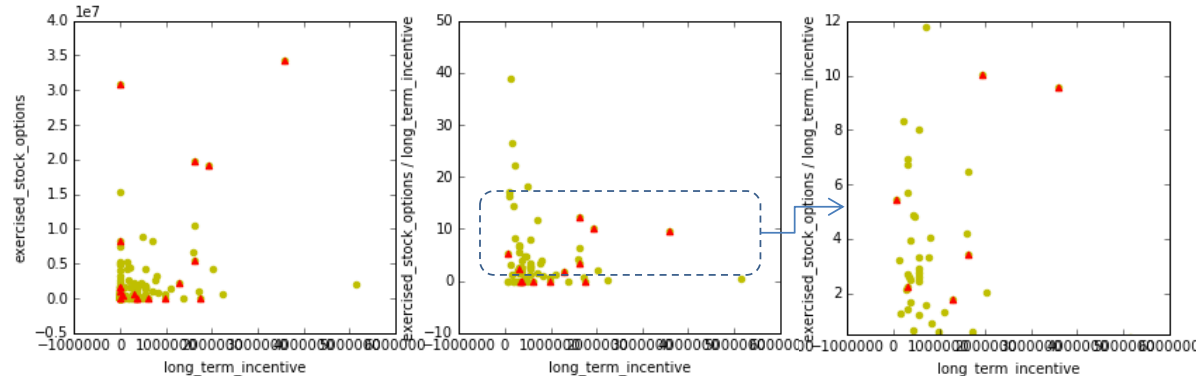


Long term incentives and exercised stock options

Those with higher long term incentives would typically not offload their options unless they are aware of insider knowledge and hence would be POIs.

On the contrary, those with limited long term incentives would especially under times of uncertainty offload the options they might have more aggressively.

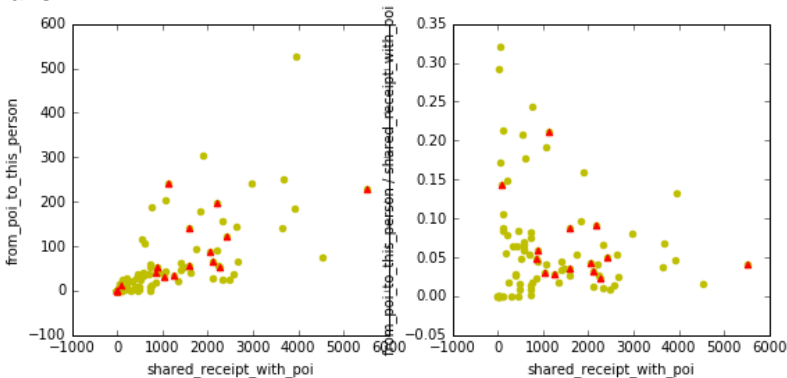
Ratio is sort of an inverse indicator



Emails from POIs to a person versus total shared emails

Given emails were the third category type I reviewed related features. But my initial inclination was to drop these altogether given the high number of NaN.

However esp shared receipt with poi showed some patterns emerging and some semblance of discrimination b/w POI and non POI



Question 3

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"]

Given I used DecisionTrees to compute the feature importance this seemed the obvious choice to start with. However to overcome some of the inherent disadvantages of decision trees mainly around high variance (as highlighted under <http://scikit-learn.org/stable/modules/tree.html>) my next choice was to use ensemble method provided by ExtraTreesClassifier (usually allows to reduce the variance of the model a bit more, at the expense of a slightly greater increase in bias). I also reviewed kNearestNeighbors classifier and SVM

	Parameters	Accuracy	Precision	Recall	F1	Time (s)
ExtraTreesClassifier	default	0.855	0.611	0.356	0.449	18.471
kNearest Neighbors	default	0.881	0.747	0.431	0.546	0.952
SVM	default	-	-	-	-	-

Though the results are above the required thresholds (0.3 for precision and recall) even using default parameters, I did review the performance of kNearestNeighbors in more detail (see Question 4)

One thing to note is the outcome of support vector classifier. Using the default parameters the error message was:
Precision or recall may be undefined due to a lack of true positive predictions.

My initial assumption was that this must be due to the fact that this algorithm calls for scaled features.

Applying linear scale (MinMaxScaler) but also adding additional feature provided an interesting performance "upgrade".

Note: I used cv = train_test_split

```
['poi', 'bonus', 'exercised_stock_options', 'totComp']
SVC(C=5000, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape=None, degree=3, gamma=20, kernel='rbf',
    max_iter=-1, probability=False, random_state=None, shrinking=True,
    tol=0.001, verbose=False)
Accuracy: 0.892 Precision: 0.5 Recall: 1.0 F1: 0.667
```

Question 4

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"]

Tuning the parameter is to ensure minimize the generalization error and achieve a balance between bias and variance and avoiding overfitting. If not done properly the algorithm will overfit and seem to be working well on training data sets but not generalized enough.

I used GridSearchCV to fine tune my classifier using kNearestNeighbors.

	Parameters	Accuracy	Precision	Recall	F1	Time (s)
KNearestNeighbors ... cv= StratifiedShuffleSplit	default values ... clf = KNeighborsClassifier() <i>KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski', metric_params=None, n_jobs=1, n_neighbors=5, p=2, weights='uniform')</i>	0.881	0.747	0.431	0.546	0.952
KNearestNeighbors ... cv= StratifiedShuffleSplit	param_grid = {'n_neighbors': [2,3,4,5,10], 'leaf_size':[20,30,40], 'weights': ['uniform','distance']} => best_estimator_: <i>KNeighborsClassifier(algorithm='auto', leaf_size=20, metric='minkowski', metric_params=None, n_jobs=1, n_neighbors=3, p=2, weights='uniform')</i>	0.869	0.707	0.369	0.525	212
kNearest Neighbors ... cv= train_test_split	param_grid = { 'n_neighbors': [3,4,5,6,7,8,9,10], 'leaf_size':[1,2,3,5,10], 'weights': ['uniform','distance']} => best_estimator_: <i>KNeighborsClassifier(algorithm='auto', leaf_size=1, metric='minkowski', metric_params=None, n_jobs=1, n_neighbors=4, p=2, weights='uniform')</i>	0.889	0.667	0.400	0.500	1

Note: as I mentioned earlier results vary and parameters value as well depending on cross validation strategy used. Interestingly the default parameters provide a better F1 score than the ones from GridSearchCV. After further review it seems that GridSearchCV averages across the different runs and hence the measures might be different. I nevertheless used the parameters from the suggested best_estimator_ as my input for my final classifier. And measures were actually better.

Question 5

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"]

Purpose of validation is to assess if the model performs well in a generalized setup. Classic mistake is not to use a representative test data but rely on training data to assess performance of the model. Also if the model has a high variance then the validation can lead to false accuracy i.e. overfitting

I cross validated using train_test_split with test_size of 30%.

Interestingly and as shown on slide 5, using cv provided in the tester module (StratifiedShuffleSplit) drove an entirely different feature list.

This is not entirely surprising since the training outcome depends on the training inputs.

Question 6

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"]

I reviewed both precision and recall. In the context of identifying POI both measures are relevant.

Precision will allow to ensure non POIs are not wrongly implicated. Conversely a low recall will suggest that POIs who potentially should be held accountable might "slip away".

My strategy was to ensure that both precision and recall are balanced.

My identifier has a decent F1 score which means it can identify POIs both reliably (recall of 0.45) and accurately (precision of 0.73).

However given the imbalance b/w precision and recall it is clear that it errs more towards missing out in identifying POIs when they should have been flagged as such i.e. higher false negatives compared to false positives.

```
Number of records: 146
Number of features: 21
Ratio of NaN in the data set: 0.44
Number of POIs in the data set: 18.0 which is 0.12
Number of records post removal of outliers: 144

#####
Shortlisted features list ...
['poi', 'exercised_stock_options', 'totComp']
#####

Final run ...
KNeighborsClassifier(algorithm='auto', leaf_size=2,
metric='minkowski',
metric_params=None, n_jobs=1, n_neighbors=3, p=2,
weights='uniform')
Accuracy: 0.88017
Precision: 0.72920 Recall: 0.44700
F1: 0.55425 F2: 0.48450
Total predictions: 12000 True positives: 894
False positives: 332 False negatives: 1106 True
negatives: 9668

training time: 1.058 s
```

My sources

- Sherron Watkins, former vice president, Enron Corporation
<https://www.youtube.com/watch?v=RSMnvCMS0W8>
- Enron fall
<http://www.slideshare.net/irwanarfandi/enron-fall>
- Scikit-learn
<http://scikit-learn.org/stable/index.html>

Final Project Rubric

Criteria	Meets expectations
Quality of Code	
Functionality	Code reflects the description in the answers to questions in the writeup. The write up clearly specifies the final analysis strategy.
Usability	poi_id.py can be run to export the dataset, list of features and algorithm, so that the final algorithm can be checked easily using tester.py.
Understanding the Dataset and Question	
Data Exploration (related mini-project: Lesson 5)	Student response addresses the most important characteristics of the dataset and uses these characteristics to inform their analysis. Important characteristics include: <ul style="list-style-type: none"> • total number of data points • allocation across classes (POI/non-POI) • number of features • are there features with many missing values? etc.
Outlier Investigation (related mini-project: Lesson 7)	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.
Optimize Feature Selection/Engineering	
Create new features (related mini-project: Lesson 11)	At least one new feature is implemented. Justification for that feature is provided in the written response, and the effect of that feature on the final algorithm performance is tested.
Intelligently select features (related mini-project: Lesson 11)	Univariate or recursive feature selection is deployed, or features are selected by hand (different combinations of features are attempted, and the performance is documented for each one). Features that are selected are reported and the number of features selected is justified. For an algorithm that supports getting the feature importances (e.g. decision tree) or feature scores (e.g. SelectKBest), those are documented as well.
Properly scale features (related mini-project: Lesson 9)	If algorithm calls for scaled features, feature scaling is deployed.
Pick and Tune an Algorithm	
Pick an algorithm (rel mini-project: Lessons 1-3)	At least 2 different algorithms are attempted and their performance is compared, with the more performant one used in the final analysis.
Tune the algorithm (mini-project: Lessons 2, 3, 13)	Response addresses what it means to perform parameter tuning and why it is important. At least one important parameter tuned, with at least 3 settings investigated systematically, or any of the following are true: <ul style="list-style-type: none"> • GridSearchCV used for parameter tuning • Several parameters tuned • Parameter tuning incorporated into algorithm selection (i.e. parameters tuned for more than one algorithm, and best algorithm-tune combination selected for final analysis)
Validate and Evaluate	
Usage of Evaluation Metrics (related mini-project: Lesson 14)	At least two appropriate metrics are used to evaluate algorithm performance (e.g. precision and recall), and the student articulates what those metrics measure in context of the project task.
Validation Strategy (related mini-project: Lesson 13)	Response addresses what validation is and why it is important. Performance of the final algorithm selected is assessed by splitting the data into training and testing sets or through the use of cross validation, noting the specific type of validation performed.
Algorithm Performance	When tester.py is used to evaluate performance, precision and recall are both at least 0.3.