# Document the work

The idea of this file is the replace the README.md in terms of documenting the work and leaving it for files description.

In this document I will follow the questions suggested and I will use “#x” numbers reference to sections in the script (poi\_id.py).

***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.***

My first objective is to get some info on the Enron case and understand what have happened to be able

to look at the available data with the right context. To do that I watched, as recommended, the TV documentary :

"Enron : the smartest guys in the rooms" . Based on that, I coul conclude the following statements about

my comprehension to help me start wrinting somo hipothesis and some answers :

1) The fraud consisted in "cooking" the accounting books to keep stock prices growing

2) Executives profits were mostly made out of seeling those overpriced stock, exercise their stock options

3) Benefited executives were those who executed those stocks at higher prices -before the crash-

4) We have available 4 features that are directly related with item #3 which are :

total\_stock\_value

exercised\_stock\_options

restricted\_stock\_deferred

restricted\_stock

where total\_stock\_value = exercised\_stock\_option + (restricted\_stock - restricted\_stock\_deferred)

Based on this I’am planning to focus on finanttial features correlating payments based features like ‘salary’ and ‘bonus’ with those stock features above mentioned. Given the fact that exercised\_stock\_options + restricted\_stock are actually summed up to account for the total ‘cash’ made based on stock by the people that has information on stock value , I will end up creating a new feature named ‘cash\_from\_stock’ to collect that.

Machine learning techniques are very usefull in this case basically because we have a dataset that includes features and labels (poi or non poi); our objective is to predict, based on some features values, if the sample (enron employee in this case) can be categorized as poi or non poi. For that reasson, since we have data already categorized, our problem is a binary (poi / nor poi), classification (discrete, no regression), supervised (train set with features and labels) machine learning problem.

***Were there any outliers in the data when you got it, and how did you handle those? [relevant rubric items: “data exploration”, “outlier investigation”]***

The actual code for this question and answers is poi\_id.py Section #1

Lets load and explore the dataset.

Let start with some of the datasets characteristics.

Total number of data points : 146

Allocation between the POI and non-POI: POI = 18, non-POI = 128

Total number of features: 21

There are missing values in the features ?: Yes, there are ‘NaN’ and/or missing values

Which features contains missing values: Most of the features has missing values. I used the pandas ‘isnull’ function and returned True for all of the features but ‘poi’ and ‘email\_address’. Cross checking looking at some data I found that original NaN in email\_address were treated incorrectly so only the ‘poi’ feature has complet information-

Features with missing values: ['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', 'director\_fees', 'deferred\_income', 'long\_term\_incentive', 'from\_poi\_to\_this\_person']

Features with no missing values: ['poi', 'email\_address']

Before working on outliers identification I will drop all non-finantial features from the dataset, drop the ‘TOTAL’ data point (summarization) and drop all data points that cointains all ‘NaN’.

To check for outliers we will look at pandas describe() statistics and will compute if max() values in the features are 2 \* std from the mean.

Almost all features has outliers. Since in the previous iteration I have reviewed (old\_poi\_id.py) LinearSVC, LogisticRegression and RandomForestClassifier and the Forest algorithm was the best performer (and I was recommended to use it) . Trees family of classifiers dont require preprocessing and provide a feature selection statistics.

Lastly, I will fill ‘NaN’ with 0 (zero) .

***What features did you end up using in your POI identifier, and what selection process did you use to pick them?***

The actual code for this question and answers is poi\_id.py Section #2

Now I have the following features in the cleaned datasets

Features remaining after cleaning mail: ['salary' 'deferral\_payments' 'total\_payments' 'exercised\_stock\_options'

'bonus' 'restricted\_stock' 'restricted\_stock\_deferred' 'total\_stock\_value'

'expenses' 'loan\_advances' 'other' 'poi' 'director\_fees' 'deferred\_income'

'long\_term\_incentive']

NaNs were filled with 0.

Before feature selection I will create the new feature mentioned earlier (see rationale above) : ‘cash\_from\_stock’ . Given the fact that total\_stock\_value is dependent on exercised\_stock\_options , restricted\_stock and restricted\_stock\_deferred I will first train a RandomForestClassifier with all the features, based on the result I will delete those features and train again with the new created feature and cross validate the results.

The new future will be created as :

my\_dataset['cash\_from\_stock'] = my\_dataset['exercised\_stock\_options'] + my\_dataset['restricted\_stock']

***Did you have to do any scaling? Why or why not?***

I wont since I will be training a RandonForestClassifier

see : <http://scikit-learn.org/stable/auto_examples/preprocessing/plot_scaling_importance.html>

see : <https://stackoverflow.com/questions/8961586/do-i-need-to-normalize-or-scale-data-for-randomforest-r-package>

***- explain what feature you tried to make, and the rationale behind 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”, “intelligently select features”, “properly scale features”]***

Before feature selection I will create the new feature mentioned earlier (see rationale above) : ‘cash\_from\_stock’ . Given the fact that total\_stock\_value is dependent on exercised\_stock\_options , restricted\_stock and restricted\_stock\_deferred I will first train a RandomForestClassifier with all the features, based on the result I will delete those features and train again with the new created feature and cross validate the results.

The new future will be created as :

my\_dataset['cash\_from\_stock'] = my\_dataset['exercised\_stock\_options'] + my\_dataset['restricted\_stock']

***What algorithm did you end up using? What other one(s) did you try?***

RandomForestClassifier – I have reviewed (old\_poi\_id.py) LinearSVC, LogisticRegression and RandomForestClassifier

***How did model performance differ between algorithms? [relevant rubric item: “pick an algorithm”]***

***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? What parameters did you tune? (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 items: “discuss parameter tuning”, “tune the algorithm”]***

***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 items: “discuss validation”, “validation strategy”]***

***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”]***