**Class**: Intro to Machine Learning

**Project**: Enron POI Detector

**Student**: Mikhail Advena

**Introduction**

This is a paper describing the final project for “Introduction to Machine Learning” course. The course was broad and presented a lot of new and challenging concepts. But the instructors were terrific and I this project represented a great opportunity for me to apply all the learnt concepts in practice.

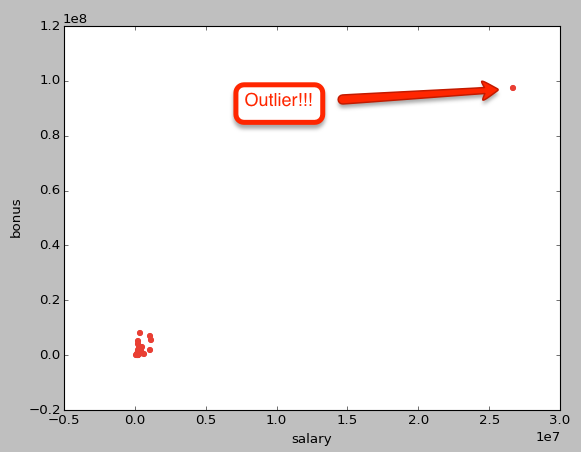
The analysis was done based on the financial data for 146 people in Enron corporation. The goal was to build a prediction on whether a person is “POI” – somebody who became a subject of investigation which followed Enron collapse.

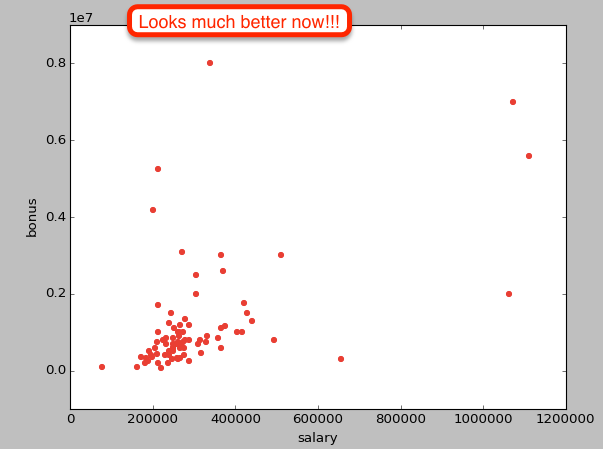
**The Enron Data: high level description and first insights**

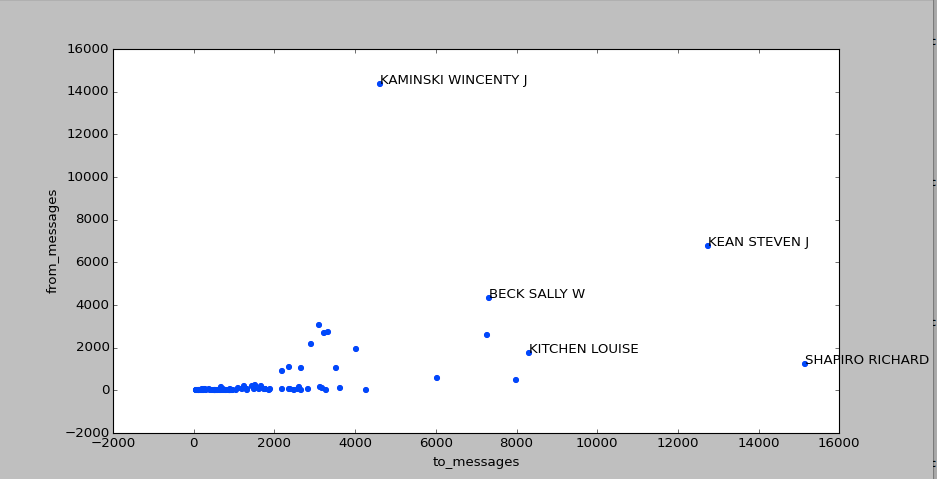
* Financial data consist of 146 records
* Out of 146 records, 18 are marked as POIs and 138 are marked as not POIs
* Because the share of POIs in the data set is small (less then 15%), Accuracy will not be the best measure to assess the efficiency of algorithm; precision and recall will be used instead
* Each record in the financial database has 21 features
* Because number of record is significantly higher that number of features, we should not worry about over fitting
* Features consist of four broad feature classes
  + POI classification: only one feature – “POI”
  + Financial: the biggest set of features which includes the characteristics like salary, bonus, total\_payment, etc
  + Email activity: to\_messages, from\_messages, from\_poi\_to\_this\_person and from\_this\_person\_to\_poi
  + Descriptive: email address, other (I do not really know what this one means)
* “POI” will be used as a label, descriptive features will not be used and I will need to use both financials and email activity features in my analysis

**Identifying and Removing Outliers**

* Visualization is always a good first step to understand the data. Below is the graph with show the distribution of Salary vs Bonus
* This outlier’s name is “TOTAL” and it need to be excluded from the dataset
* Another graph show the data after removing “TOTAL”
* Another visualization shows the distribution of emails sent vs received (in BLUE). Some people clearly wrote / received more emails then others, but after adding names of the people with most emails sent / received I did not find any “anomalies”

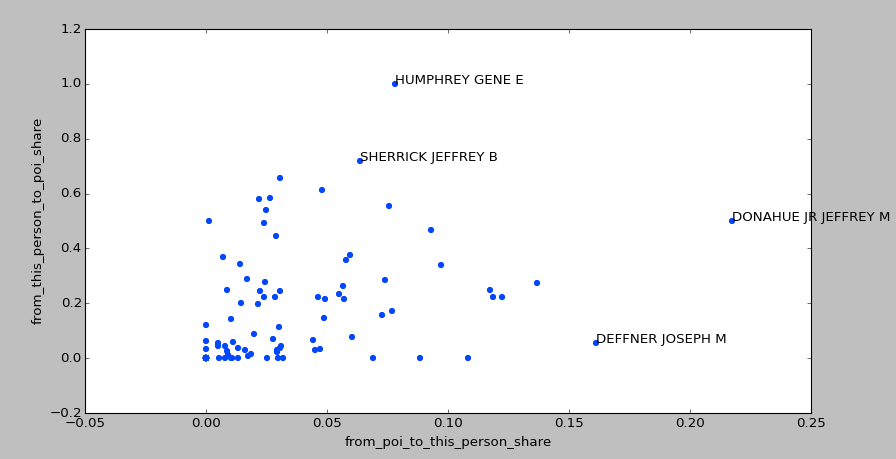






**Create New Features**

* Number of emails sent to / received from POI can be important prediction if a person is POI herself. One problem with these numbers is that some people write / receive way to many emails than others (as indicated on the graph above).
* One way to address this issue would be to introduce two new features, which look at relative share of emails to / from POI
  + from\_this\_person\_to\_poi\_share = from\_this\_person\_to\_poi / from\_messages
  + from\_poi\_to\_this\_person\_share = from\_poi\_to\_this\_person / 'to\_messages'
* As we can see from the graph below, these new features have more “reasonable” distribution



**Algorithm Selection and Tuning**

I used two algorithms: Decision Tree Classifier and SVC

**Support Vector Machines Classifier (def SVM)**

Feature Selection

* SelectKBest feature selection was deployed to select eight best features
* Chi2 score function could not be used because of negative data in the feature set
* That is why f\_classif was the only option
* Below is the resulting list of 8 best features

Best features = ['salary', 'bonus', 'deferred\_income', 'total\_stock\_value', 'exercised\_stock\_options', 'long\_term\_incentive', 'restricted\_stock', 'from\_this\_person\_to\_poi\_share']

Feature Scaling

* Support Vector Machine classifier Is sensitive to the “absolute distance” so the features were normalized using sklearn.preprocessing.normalize

Algorithm tuning

* I used GridSearchCV to find the best set of parameters; the following range of parameters was tested
  + ' kernel' : ('rbf', 'linear', 'poly'),
  + 'C' : [1, 2, 4, 6]}
* Here is one set of optimal parameters and corresponding algorithm results
  + Best parameters are {'kernel': 'rbf', 'C': 1}, 'min\_samples\_leaf': 2}
  + Accuracy = 0.886363636364
    precision = 0.0
    recall = 0.0

Conclusion

* Support Vector Machine classifier seems to have strong bias toward labeling everybody as “not POI”
* I do not know how deal with this and decided to try another classifier instead

**Decision Tree Classifier (def DecisionTree)**

Feature Scaling

* Decision Tree Classifier does not require feature scaling so it was not performed

Feature Selection

* This algorithm allows calculation of the feature importance. That is why I started from using all Financial and Email statistics, calculated feature importance and looked at how algorithm predictive power changes when top 9 and top 5 features are used
* Here is the algorithm results when all features were used
  + Features importance

salary 0.0

deferral\_payments 0.0

total\_payments 0.0589569160998

loan\_advances 0.0

bonus 0.0

restricted\_stock\_deferred 0.0

deferred\_income 0.0

total\_stock\_value 0.0555777866702

expenses 0.119345984008

exercised\_stock\_options 0.250793650794

other 0.190136054422

long\_term\_incentive 0.0

restricted\_stock 0.0704993108354

director\_fees 0.0

to\_messages 0.0

from\_poi\_to\_this\_person 0.0

from\_messages 0.0

from\_this\_person\_to\_poi 0.0

shared\_receipt\_with\_poi 0.118635875403

from\_this\_person\_to\_poi\_share 0.136054421769

from\_poi\_to\_this\_person\_share 0.0

* + Predictive power

Accuracy: 0.82087 Precision: 0.32248 Recall: 0.31200 F1: 0.31715 F2: 0.31404

* In second iteration I used the nine features with featiure\_importance > 0 plus from\_poi\_to\_this\_person\_share . Here is the corresponding feature importance and algorithm results
  + Feature importance

total\_payments 0.0846560846561

total\_stock\_value 0.0

expenses 0.170139634801

exercised\_stock\_options 0.272206660442

other 0.130612244898

restricted\_stock 0.0281712685074

shared\_receipt\_with\_poi 0.178159684927

from\_this\_person\_to\_poi\_share 0.136054421769

from\_poi\_to\_this\_person\_share 0.0

* + Predictive power

Accuracy: 0.84273 Precision: 0.40427 Recall: 0.37900 F1: 0.39123 F2: 0.38380

* In the last iteration I left only 5 most important features. Here is the corresponding feature importance and algorithm results
  + Feature importance

expenses 0.274138896159

exercised\_stock\_options 0.199176954733

other 0.273713974654

shared\_receipt\_with\_poi 0.119248664015

from\_this\_person\_to\_poi\_share 0.133721510439

* + Predictive power

Accuracy: 0.84243 Precision: 0.44539 Recall: 0.42000 F1: 0.43232 F2: 0.42484

* Algorithm performance is the best when top 5 features are used. I will use this set of features for the algorithm tweaking

Algorithm tuning

* Decision Tree Classifier has a set of parameters and tuning them is important in order to optimize its performance. It is similar to driving a vehicle at 65mph at highway vs. only 20mph on small and bumpy road. You need to use appropriate speed (parameters) to get to your destination fast (get higher predictive power
* I used GridSearchCV to find the best set of parameters; the following range of parameters was tested
  + 'criterion' : ('gini', 'entropy'),
  + 'min\_samples\_split' : [2,4,6],
  + 'min\_samples\_leaf' : [1,2,3]
* Optimal parameter set is not consistent, but algorithm predictive power is consistently better vs. when default setting are used
* Here is one set of optimal parameters and corresponding algorithm results

Best parameters are {'min\_samples\_split': 4, 'criterion': 'entropy', 'min\_samples\_leaf': 2}

Accuracy: 0.85921 Precision: 0.50875 Recall: 0.42150 F1: 0.46103 F2: 0.43647

**Analytics Validation**

Why validation is important

* Validation is an important process which ensures that deployed algorithm will have predictive power on new data
* If we use same data set for the algorithm training and assessment of its accuracy we can “overfit” the algorithm to this data set and it will lack predictive power on new data
* The edge case would be making algorithm to “remember” all the features / label of the training data set. Such algorithm would have 100% accuracy on this data set, but “would not know what to do” when a new data point arises

Validation process and results

* I used train / test split to validate my algorithm (**def validate**)
* sklearn.cross\_validation.train\_test\_split function was used to perform the split
* I used 70% of data for training and 30% of data for testing
* The results were higher than that produced by Udacity test\_classifier

**Performance**

Metrics used to access performance

* Out of 146 records, 18 are marked as POIs and 138 are marked as not POIs
* It is strong indication that share of POIs in the data set is small (less then 15%).
* This means that Accuracy (share of correctly predicted outcomes) will not be the best measure to assess the efficiency of algorithm. For example, the algorithm which always predict “not POI” will have an accuracy of approximately 100% - 15% = 85%, which is pretty high, at the same time all POIs will be predicted inaccurately
* Precision and Recall from another hand will be much better metrics to access accuracy of identifying POIs. Recall measures what percent of “true POIs” was identified, Precision measures what percent of rightly predicted POIs

Results

* As discussed in the “Decision Tree Classifier” section, its algorithm (after selecting the most important features and tuning algorithm parameters) consistently delivers both Precision and Recall of > 0.4

**Discussion and Conclusions**

* It was very fun to work on this project. I felt like I learnt a lot and managed to put into practice all the cool concepts learned during the course
* Precision / Recall of only ~40% does not look like a lot. It is better than “random” picking as the share of POI in the data set is only ~15%, but is far below of “theoretically possible” 100%
* Key reasons:
  + Small data set: only 146 records, only 18 POIs
  + Limited set of used features: while financials can say a lot in this analysis, there are clearly a lot of people who made good money at Enron but did not become POIs (whether they were innocent or prosecution was not able to build a case against them is another question)
* Potential steps to improve the analysis:
  + Get more data records
  + Use emails to enrich feature set
* I am really looking forward applying my new skills on other projects!!!