Enron Submission Free-Response Questions

By Lieby Cardoso
Udacity – Data Analyst Nanodegree

1. 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. 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 energy company Enron enacted bankruptcy after executing several accounting and tax frauds. The purpose of this project is to use data analysis and machine learning techniques like features selection, preprossessing and evaluating models to predict if an employee has been involved with fraud. Those that were pointed out are called POIs (Person of Interest).

The data set used is composed of financial information, the emails sent and received by the people of interest (POI) and the identifier of whether the key person is a POI. Jeffrey Skilling (CEO) and Kenneth Lay (chairman and CEO) have many above average values, but justified by the importance of their positions.

The dataset attributes such as salary, bonuses, actions, incoming and outgoing e-mail numbers were investigated and cleaned up.

When I was looking for outliers I found a POI called TOTAL. It was an outlier whose content is the sum of the other values, we deleted this record. The second largest value receive was Kenneth Lay, but, as you know this make perfect sense since he was the CEO and chairman of Enron.

The record "THE TRAVEL AGENCY IN THE PARK" has been deleted. Although "THE TRAVEL AGENCY IN THE PARK" had received $ 350,000 in payments 2 days before Enron's bankruptcy and Sharon Lay (sister of Kenneth Lay) owned 50% of the company, this record will not be considered a POI.

I found other persons in the dataset that need our analysis. "LOCKHART EUGENE E" doesn't have any value assign. "CHAN RONNIE" had stock and income put off to a later time causing a total payments equal a 0, and he had no message sent or received. Since none of them are POI, they presence on the data set is not justified.

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 importance's 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"]

After clearing the data, I created two new features:

1. 'message_poi_ratio': percentage of emails related to the POI, both sent and received;
2. 'message_others_ratio': messages not related with a POI.

After this change, I deleted all variables email related, such as: 'email_address', 'from_messages', 'from_poi_to_this_person', 'from_this_person_to_poi', 'shared_receipt_with_poi', 'to_messages'.

Once the dataset was ready the first thing I did was apply the pandas.DataFrame's corr() function on all variables to get the correlation between them. After that I chose not to use the variables [total_payments', 'total_stock_value'] because they are the sum of other variables in the dataset.

To get a suitable number of variables to be used I used the SelectKBest function with:

1) Anova stats on a raw version of the data

2) Chi-2 stats with a standardize (MinMaxScaler) version of data

To see how I clean the data go to:
https://github.com/liebycardoso/ML_Enron/blob/master/Enron_Data_Analysis.html

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"]

The answer I was pursuing to know, based on my data, was if a given person is a POI or not. Since there are only two possible answers, we can say that this is a classification task and to help me define the way forward, I have tested multiple classifiers and their default parameters.

Model	Precision	Recall	Accuracy
KNeighbors	0.581633	0.0570	0.868800
RandomForest	0.460692	0.1465	0.863333

ExtraTreesClassifier	0.423295	0.1490	0.859467
AdaBoost	0.418319	0.3060	0.850733
NearestCentroid	0.376596	0.2655	0.843467
GradientBoostingClassifier	0.263557	0.2260	0.812600
DecisionTree	0.257143	0.2475	0.804333
LogisticRegression	0.164218	0.1900	0.763067
Naive Bayes	0.219135	0.6230	0.653733

Based on the accuracy, precision and recall value, I chose AdaBoost, GradientBoostingClassifier and NearestCentroid to improve performance. My goal was to achieve a precision and recall value higher than 0.3. In this scenario, with AdaBoost I achieve my goals even with default parameters.

Scores obtained after tuning:

Model	**Precision**	**Recall**	**Accuracy**
KNeighbors	0.39456	0.37700	0.83980
RandomForest	0.43103	0.13750	0.86080
ExtraTreesClassifier	0.22082	0.68300	0.63640
AdaBoost	0.36377	0.63950	0.80280
NearestCentroid	0.35733	0.61550	0.80113
GradientBoostingClassifier	0.31915	0.23250	0.83153
DecisionTree	0.26790	0.75200	0.69293
LogisticRegression	0.30177	0.57200	0.76647

At some point in the Project I was curious about how much I could improve other classifiers, so I continued the tuning process. Only Gaussian Naive Bayes was left out, because this estimator does not have parameters to work with, just priors.

I got three models that stood out, both KNeighbors and NearestCentroid had a superior performance when compared to others, but the chosen one was AdaBoost which had precision values like KNeighbors and NearestCentroid, but a higher recall value.

To see how I've decide for this classifier, check:
https://github.com/liebycardoso/ML_Enron/blob/master/Enron_Model_Selection.html

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”]

With the gridsearcv function an exhaustive search of the best parameter for the estimator is done, sometimes in this search for the best model you set a value that increase the Recall metric and decrease Precision. To be successful in this task the range of parameters must be carefully selected, in my test I had cases where my selection of parameters was incorrect and the tuning model returned metrics worse than its version with default parameters.

With only 18 POI records in the data set, there are few examples for learning. I tried reducing the dimensionality of the data with PCA and selecting the features with SelectkBest, but in some cases, there was no improvement as I would like. I used MinMaxScale to transforms features by scaling then to a range between 0 and 1.

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”]

To ensure that our data model can predict results, even in an unknown dataset, we need to separate our data sample into a training set and a test set. Not doing this division can lead us to an overfitting situation, which is when the model foresees perfectly what has already seen, but fails to predict what has not been tested.

While I was testing the models, I split the dataset using cross_validation.train_test_split and set the test set size in 30%. In a dataset, so small it is interesting to use StratifiedShuffleSplit as a cross validator, with it the process of fitting and training is repeated 1000 times. I used this validator in the final test (Script tester.py).

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”]

Once a model is considered ready it is important to choose the statistical metrics that can help us estimate how accurately and precise our model is. I started this project using the F1 metric as a reference, but it is a balance between precision and recall and it ended up returning a high value when one of the measurements was high and I needed both recall and precision to be over 0.3. This project focused on precision, recall and accuracy.

The model chosen after tuning was AdaBoostClassifier that returned the most balanced values of precision, recall and accuracy.

Achieved values:

Classifier	Precision	Recall	Accuracy
AdaBoostClassifier before tuning	0.418319	0.3060	0.850733
AdaBoostClassifier after tuning	0.36377	0.63950	0.80280

As you may realize after tuning the model has lost a bit of accuracy and precision, but has achieved a significant increase in recall value. Which is quite significant for me because it tells me that the model has greatly improved its ability to properly point a POI.

The performance metrics of our model tells us that it is capable of:

- For POI records predicted, 36% of them are POIs (Precision)
- Among all available records it can identify 64% of POIs (Recall)
- Correctly predict 80% of the values (Accuracy)

Identifying whether a particular employee is related to a fraud scheme is a very important task and that if misclassified can have profound consequences for the people listed. For this type of project would be very important to have a dataset with more than 18 POIs for training.