P5- Machine Learning Project – Identify Fraud from Enron Email

Edmund Wong

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 goal of this project is to find a machine learning algorithm along with a set of features that is able to effectively predict whether an employee of Enron, a large company in the early 2000s that collapsed due to corporate fraud, is considered a person of interest (POI) or not. Much of the data became public record, including both financial and email data. The dataset used for this project includes 146 employees at Enron. There are total of 20 features that could be used for this project (14 financial related, 6 email related). There is an allocation of 18 POIs and 128 non-POIs in the dataset. There were a variable number of missing values for each feature (identified as "NaN") in the dataset. Here is the breakdown of all the features that had missing values and their respective "NaN" count:

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

To look for outliers, I generated a CSV file for the dataset so I could be able to visually observe the dataset. Some executives were making much higher salaries and bonuses that can be considered outliers, but it made sense not to remove them as it is likely that those were their actual salaries/bonuses. The obvious records for me to remove were 'THE TRAVEL AGENCY IN THE PARK' (which was not a person), 'TOTAL' (which was the sum of all the employees), and 'LOCKHART EUGENE E' (which had no data at all).

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"]

The features that I ended up using for my POI identifier are: salary, bonus, total_stock_value, exercised_stock_options, and ratio_to_poi_email (SelectKBest scores were: 10.88, 15.82, 23.43, 27.17, 12.82, respectively). First, I decided to avoid using features where more than 50% of its records had missing values. I also avoided using email_address feature, which did not seem like a good feature to use for prediction. For the remaining features that I did not avoid, I decided to impute the missing values of these features using the median. I also created two new features: ratio_from_poi_email and ratio_to_poi_email. It makes sense to know what percentage of their incoming emails were from POIs (and also what percentage of their outgoing emails were to POIs).

My features were selected as the best features determined by SelectKBest, which removes all but the k highest scoring features. I used a balanced value of k=5 because I wanted to avoid high bias (from using too few features) and high variance (from using too many features).

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 algorithm I ended up using was GaussianNB, as it performed the best. The other algorithm that I have tried is SVC.

GaussianNB:
Accuracy: 0.86407
Precision: 0.48618
Recall: 0.34300
F1: 0.40223

SVC:
Accuracy: 0.88120
Precision: 0.67357
Recall: 0.21150
F1: 0.32192

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 parameters of an algorithm means to tweak the model that will optimize the performance of the model. If this is not done, then the performance will not be be optimal and will result in possibly lower evaluation metrics. For example, tuning the C parameter in SVC incorrectly could either cause high misclassification of datapoints or the opposite (overfitting). C is the penalty parameter for fitting of the kernel function. I selected a wide range of C values to see which value is optimal for this dataset. I used GridSearchCV to fit using each possible choice of parameters to determine the best estimator.

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"]

Validation is a strategy that will ensure that each record in the dataset will have the same probability to occur in both training and testing sets. This will prevent a classic mistake where the model will overfit to only one training set. I used StratifiedShuffleSplit to cross-validate my analysis. The dataset is randomly split into fixed proportions into training and testing sets over many iterations. I used n_iter=10 adnd test_size=0.1, which performs 10 re-shuffling and splitting iterations where the test set is 10% of the entire dataset.

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"]

Two evaluation metrics is precision and recall, where my algorithm's average performance is 0.48618 and 0.34300, respectively. Precision tells us that of all the employees that we identified as POIs, what is the percentage of them that are actually POIs? Recall tells us that of all the POIs in our dataset, what is the percentage of them that are correctly identified? Both metrics are important to the overall performance of the model.