Kathleen Cravotta December 2016

|  |  |
| --- | --- |
| **Let me know if I’m missing something about how to organize this project. I take everything literally, and these are literally different requirements.** | |
| Description from Project Submission Page: | Feedback from Round 1 Submission: |
| Document the work you've done by answering (in about a paragraph each) the questions found [**here**](https://docs.google.com/document/d/1NDgi1PrNJP7WTbfSUuRUnz8yzs5nGVTSzpO7oeNTEWA/pub?embedded=true). …  Please answer each question; your answers should be about 1-2 paragraphs per question. If you find yourself writing much more than that, take a step back and see if you can simplify your response! | In the written response, ideally every step/choice of parameters and algorithm selection, tuning or feature selection should be followed by a table with the adequate figures reporting the impact of that specific step/choice on the chosen performance metrics and a consequent rationale for the choices made in the light of those results. Doing so would help the reader to properly understand the decision process. It is very important, in every scientific work, to explain and justify every step/choice as thoroughly as possible, no matter how obvious to you the various steps might look. |

Enron Submission Free-Response Questions

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 the project is to find patterns in the data that may be used to signal whether or not a person was involved in fraud at Enron. The data used in my code has financial data, such as bonus and salary, as well as summary email data, such as the number of emails from that person. This data was compiled by Katie or her colleagues using publicly available financial and email information of Enron employees or contractors.

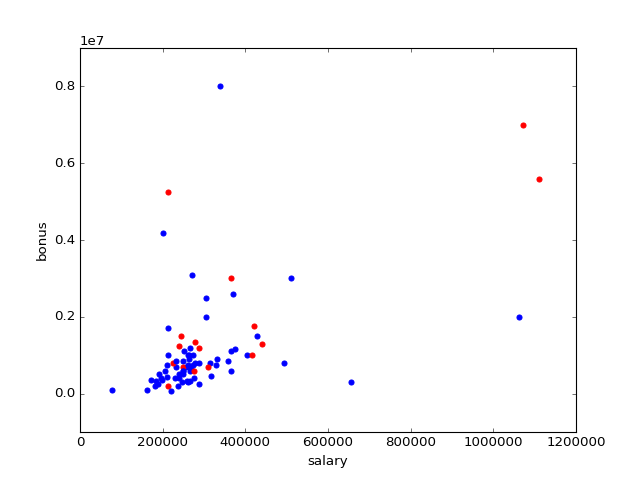
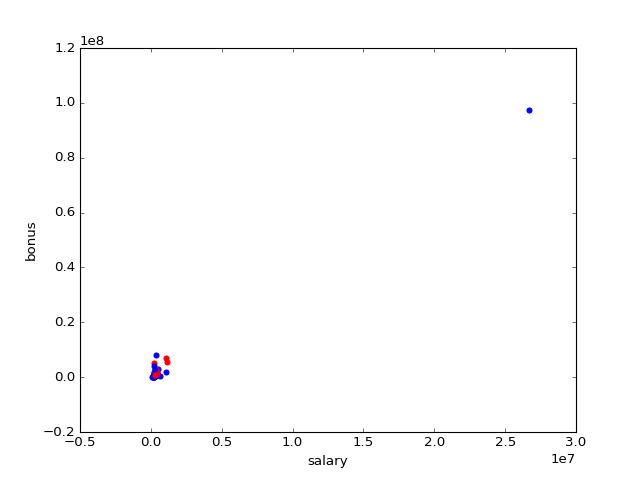
Number of records: 145

Number of poi: 18

Number of non-poi: 127

Number of Features: 21

I explored the data using 2-D visuals and max check to see if there were any outliers. The financial outlier ‘TOTAL’ popped up, which we discovered during the lessons. I removed it from the working dictionary before formatting the features for the machine learning algorithms. Here are the before and after plots of salary and bonus, where the first contains the outlier.



I continued to explore 2D plots without prominent discovery. I then took note of missing values in the data, signified by ‘NaN’. I saw that there was very little data for loan\_advances, restricted\_stock\_deferred, and director\_fees.

Number of missing data per feature

59 - to\_messages

107 - deferral\_payments

64 - bonus

20 - total\_stock\_value

51 - expenses

59 - from\_this\_person\_to\_poi

0 - poi

97 - deferred\_income

34 - email\_address

59 - from\_poi\_to\_this\_person

51 - salary

21 - total\_payments

80 - long\_term\_incentive

142 - loan\_advances

36 - restricted\_stock

128 - restricted\_stock\_deferred

59 - shared\_receipt\_with\_poi

44 - exercised\_stock\_options

59 - from\_messages

53 - other

129 - director\_fees

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”]

I created 3 parameters based on the existing data. Since the number of emails per person could be on different scales, I used a ratio in place of the number of emails to and from a person of interest so that these would be in balance with the person’s email activity. I also used ratio in place of exercised stock options, because that activity would be bound by the individual’s held stock value. Ultimately the ratio for exercised stock options didn’t seem to be used by the classifiers.

The SVM classifiers did not work, so I tried a scaler on the features for them but it didn’t help. I did also use scaled values for testing K-neighbors algorithm.

In my initial exploration, I used all the features (except email address) to get a feel for how to configure the algorithms, how was the performance, and what was feasible. I also checked the importance of features in the AdaBoost and GridSearch Decision Tree.

|  |  |
| --- | --- |
| DT Importance:  to\_messages : 0.0  email\_from\_poi\_ratio : 0.0  restricted\_stock\_deferred : 0.0  deferral\_payments : 0.0  bonus : 0.0  total\_stock\_value : 0.0746007468271  expenses : 0.208956631796  from\_this\_person\_to\_poi : 0.0  deferred\_income : 0.0987801021886  shared\_receipt\_with\_poi : 0.0  long\_term\_incentive : 0.0  salary : 0.0  total\_payments : 0.0  loan\_advances : 0.0  restricted\_stock : 0.0  email\_to\_poi\_ratio : 0.0  from\_poi\_to\_this\_person : 0.0  exercised\_stock\_options : 0.0  from\_messages : 0.0372618896376  other : 0.580400629551  exer\_stock\_ratio : 0.0  director\_fees : 0.0 | AB Importance:  to\_messages : 0.0197593425823  email\_from\_poi\_ratio : 0.0277077693591  restricted\_stock\_deferred : 0.0  deferral\_payments : 0.0247863342737  bonus : 0.0402258590179  total\_stock\_value : 0.0700948931873  expenses : 0.0957178645192  from\_this\_person\_to\_poi : 0.0391803527522  deferred\_income : 0.0225017147503  shared\_receipt\_with\_poi : 0.0224070400598  long\_term\_incentive : 0.0242565783808  salary : 0.113657959528  total\_payments : 0.0519006594209  loan\_advances : 0.0  restricted\_stock : 0.0335747421505  email\_to\_poi\_ratio : 0.0553214705143  from\_poi\_to\_this\_person : 0.0386487187755  exercised\_stock\_options : 0.245867435538  from\_messages : 0.0  other : 0.0558384513263  exer\_stock\_ratio : 0.0185528138636  director\_fees : 0.0 |

Then I ran a series of k-fold tests using different sets of features. First, I used all the features. Then I tried using the features I noted during my initial exploration and features based on importance above.

all of the features:

'to\_messages', 'email\_from\_poi\_ratio', 'restricted\_stock\_deferred', 'deferral\_payments', 'bonus', 'total\_stock\_value', 'expenses', 'from\_this\_person\_to\_poi', 'deferred\_income', 'shared\_receipt\_with\_poi', 'long\_term\_incentive', 'salary', 'total\_payments', 'loan\_advances', 'restricted\_stock', 'email\_to\_poi\_ratio', 'from\_poi\_to\_this\_person', 'exercised\_stock\_options', 'from\_messages', 'other', 'exer\_stock\_ratio', 'director\_fees'

features based on exploration intuition:

'salary', 'bonus', 'total\_payments', 'exercised\_stock\_options', 'shared\_receipt\_with\_poi', 'expenses', 'email\_to\_poi\_ratio', 'email\_from\_poi\_ratio', 'exer\_stock\_ratio'

features based on example DT importance:

'other', 'expenses', 'total\_stock\_value', 'exercised\_stock\_options', 'long\_term\_incentive', 'from\_this\_person\_to\_poi', 'from\_messages', 'restricted\_stock'

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”]

I used a grid search with a decision tree varying the depth and min samples split. It worked better than the default decision tree, and slightly better than the ada boost decision tree. It overfit. K-neighbors and Naïve Bayes did well but had less success. There was an issue with the SVM and it didn’t classify at all.

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”]

The parameter tuning can help to prevent over-fitting with high variance, and underfitting with high bias. I used a grid search varying the depth and min samples split to see if I could find better parameters, but it was still hard to find a good fit for the data. I also ran some of the parameters manually so that I could set up different test sets. I thought about if the grid search I used was trying to find a good solution, so I specified the scoring solution to ‘f1’. This ties in to validation below.

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 can be wrong when using the same data that trained the classifier, to test the classifier. When the classifier over-fits to the training data, it will appear to have high accuracy. Another thing we are looking at when validating our Enron data set is that the occurrence of positive nodes is sparse. In those cases, only looking at accuracy will make the algorithm appear to have good performance when it frequently returns negative.

To do good validations, separate data should be used to test the algorithm, that wasn’t used to train the algorithm. I used a fixed split of data as I explored different classifiers, and then I used k-fold testing to get a closer look at the performance of classifiers. When I looked at the performance, instead of looking at accuracy, I focused on Precision and Recall, below.

6. Give at least 2 evaluation metrics and your average performance for each of them. Explain an interpretation of your metrics that says something humanunderstandable about your algorithm’s performance. [relevant rubric item: “usage of evaluation metrics”]

Precision and Recall are especially important metrics for the sparse POI classifier. Precision is measures how likely the classifier is to be correct when it identifies a person of interest. My decision tree is on average 34% correct when it identifies a person of interest. Recall is how likely the classifier is to correctly identify a person of interest. My Grid Search DT finds on average 30% of the persons of interest it sees.