Neil Amin

Enron Question Responses

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 Enron was large corporation that was part of a large corruption investigation. The goal of this project is to identify potential people of interest from the emails of the employees using machine learning to find patterns. There are 146 potential candidates for corruption and we are given 21 features to use to help us find which of the these are POIs. There are 18 POIs in this dataset. We moved the outliers of Total and the NaNs.**

1. 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”, “intelligently select features”, “properly scale features”]

**I chose to create the fraction\_to\_poi and fraction\_from\_poi features because I felt those who had the most communication with POIs were most likely to be POIs themselves.**

**I computed the coefficients of each feature using decision tree importances and ranked them which helped me to decide which ones to use. I used the features with the top 4 coefficients which were salary, bonus, fraction\_to\_poi, and fraction\_from\_poi because those gave me the best precision/recall after iterative trial and error. The importances returned as follows:**

Feature Ranking:

1 feature salary (0.331377808033)

2 feature bonus (0.247721151492)

3 feature fraction\_from\_poi (0.127535296381)

4 feature fraction\_to\_poi (0.124537409493)

5 feature deferral\_payments (0.0753744043567)

6 feature total\_payments (0.0475663716814)

7 feature loan\_advances (0.0324377569154)

8 feature restricted\_stock\_deferred (0.0134498016478)

9 feature deferred\_income (0.0)

10 feature total\_stock\_value (0.0)

11 feature expenses (0.0)

12 feature exercised\_stock\_options (0.0)

13 feature long\_term\_incentive (0.0)

14 feature shared\_receipt\_with\_poi (0.0)

15 feature restricted\_stock (0.0)

16 feature director\_fees (0.0)

**That combination gave me a precision of .31635 and recall of .31050. Other combinations I used were 16 features with a precision of .30761 and recall of .29500, the top 8 features which gave a precision of .25110 and recall of .28550, the top 5 features which gave a precision of .28833 and recall of .29900, and the top 3 features which had a precision of .22593 and recall of .2300.**

**I did not use any scaling when I used decision tree because it was not required. I also did not use any scaling for Naïve Bayes either. Alternatively, SVM would require feature scaling because it depends on the distance between features.**

1. 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 tried Decision Tree and NB. Decision tree had the best performance with a precision of .32168 and recall of .31750 so I decided to use that. NB had a precision of .39315 with a recall of .16650 with the same features.**

1. 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”]

**One can tune an algorithm by giving it different parameters to change the performance of the algorithm. If tuning is not done well we will get lower performance due to over/under fit to the training data. I tuned the max features by trying square root and log 2. I also tried to set the min samples split to 2. My results for log2 were a precision of .33907 and recall of .30550. My results for square root were a precision of .33950 and recall of .31200. My results for a min samples split of 2 were a precision of .32199 and recall of .31700. My results for a min sample split of 10 were a precision of .36179 and recall of .23100. My results for a min sample split of 5 were a precision of .32818 and a recall of .28650. The results for square root and min sample of 2 were the best so I decided to go with that.**

1. 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”]

**Validation is the verification that the model vs data not used in training. Many people forget to split the data between training and testing which will result in bad performance. I used Stratified Shuffle Split to split by data: 90% for training and 10% for testing. I chose this because of the imbalance in the data which may have resulted in either the training or test set not having an POIs. The dataset is small and skewed towards non-POI, we need a technique that accounts for that or the risk is that we would not be able to assess, in the validation phase, the real potential of our algorithm in terms of performance metrics. The chance of randomly splitting skewed and non-representative validation sub-sets could be high, therefore the need to use stratification (preservation of the percentage of samples for each class) to achieve robustness in a dataset with the aforementioned limitations.**

1. 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”]

**My data was evaluated based on both accuracy, precision, and recall. I chose to use the Decision Tree because it had the best for both. Accuracy is the rate at which POI are correctly identified. Precision is the percent of employees identified as POI that are actually POI. Recall is the number of correctly identified POIs over the number of actual POIs.**