Results of Finding Enron POI

Sunda Gerard

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 use machine learning to create a powerful algorithm to identify persons of interest (POIs) in the Enron dataset. The Enron scandal brought one of the world's largest companies to bankruptcy in the early 2000's. Using a database of emails, stock options, salaries, and other now public information, the algorithm is designed to find those individuals most responsible for the downfall of Enron, known as POIs. The main idea behind the project was to find the best algorithm to identify those POIs.

There were a few outliers in the dataset that needed to be removed. There were 146 employees in the dataset. Of those 146 employees, two were identified as non-employee names. These two were "Total" and "The Travel Agency In The Park". Those were removed with the .pop command. We then identified 18 individuals as POIs. Using nan_counts, we found that one employee named "Eugene E Lockhart" had all NaNs as values in dataset fields, so he was removed from the dataset as an outlier. We also found a few individuals with greater salary and bonuses than most others, which put them as outliers, but were also POIs, so they stayed in the dataset.

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", "intelligently select features", "properly scale features"]

I compared using two algorithms to select the best features of the dataset to use. Between DecisionTree and SelectKBest, DecisionTree produced the more accurate features, so that is what I chose going forward. The ten best features identified by DecisionTree were: poi, deferral_payments, salary, from_poi_to_this_person, to_messages, exercised_stock_options, total_payments, restricted_stock, from_this_person_to_poi, and shared_receipt_with_poi. I tried some scaling, but ultimately it did not affect the accuracy, so I kept the features at 10. I engineered a new feature known as fraction_poi_emails that took the total POI emails for each employee and divided that number by their total number of individual emails. I then tested to make sure that it worked on a random employee, in this case David W Delainey. It came back that he had 10.9% of his emails to or from POIs. I was more curious about the breakdown and percentages for individual employees, so I played around with the bigger employees such as Skilling and Lay to see what their percentages were as well. I ultimately decided to forego using this new feature in the final analysis and testing stage, as other email features were already being used and I didn't want overlap. In evaluating the features selected by DecisionTree and why they are important, poi obviously is needed to identify persons of interest. Deferral_payments could indicate those individuals making lots of money, while salary does indicate those making lots of money at Enron. From_poi_to_this_person and from_this_person_to_poi show the amount of communication between the individual and POIs. Shared_receipt_with_poi could indicate that an individual was involved in a long email conversation with other prominent Enron employees. Exercised_stock_options and restricted_stock features could indicate how much Enron stock an employee had, which could indicate a POI if those features are of a high amount. Total_payments could indicate the cash-out value of employees, which large amounts could indicate a POI.

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 ended up picking DecisionTree as the algorithm to use due to best performance. I compared that algorithm with AdaBoost and LogisticRegression. DecisionTree garnered slightly higher accuracy scores than the other two at around 74% consistently. LogisticRegression came in second at 72% and AdaBoost came in last at 70%. Precision on DecisionTree was also the highest of the algorithms tried at just over 22%. LogisticRegression followed closely behind with close to 17%, while AdaBoost was just under 10% precision. Recall was highest in both DecisionTree and LogisticRegression at 25%, while AdaBoost clocked in at 12.5%. F1 score in DecisionTree was highest with 21%, followed by LogisticRegression at 20%, and AdaBoost at 10.5%.

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? 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"]

Tuning the parameters of the algorithm is making your algorithm more accurate and precise. If you don't do this well, then your performance scores on those metrics will be affected and your algorithm performance will suffer as a result. I tuned the parameters according to the results of the GridSearchCV being used. I attempted to tune both the DecisionTree and

LogisticRegression and did so with GridSearchCV to display the best parameters to use to tune the algorithms. Since I ultimately chose DecisionTree as the best algorithm, I will discuss those tuning parameters.

I tuned the min_samples_split parameter, which could range from 2 to 8, with GridSearchCv finding that 6 was optimal. I also tuned the criterion, which was between gini and entropy, with gini being optimal. Next, I looked at max_depth between 2 and 8, with 7 being optimal. Lastly, I examined the parameter of max_features with a range between 3 and 10, with 3 being optimal.

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 items: "discuss validation", "validation strategy"]

Validation is used to measure the effectiveness of an algorithm. A classic mistake if it is done wrong is overfitting the data by testing and training your algorithm on the same set of data each time. I used StratifiedShuffleSplit (sss) to validate the data. The parameters of sss used were labels of 200, test size of 0.1, and random state of 42. Precision, Recall, Accuracy, F1 and F2 scores were used to determine the effectiveness of the sss. We will also be able to use sss to help find the most optimal parameters for training and testing our algorithm.

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"]

The main focused metrics I used were precision, recall, and accuracy. I was able to achieve different results each time due to the variance of the data tested, but consistently received scores above .3. My precision was usually around a .85 level with the tested data. Precision indicates true positives or those employees who are POIs are actually POIs. This means that our data was usually accurate in predicting those individuals who were POIs.

The second metric I used was recall. This metric was consistently around .44 to .45. Recall indicates false negatives or those individuals who are actually POIs who are not identified as a POI in the algorithm. I would have like for this number to be better, but it is still above the .3 threshold given by the project rubric. The accuracy was fairly high at around .91, although this could be not as reliable a metric due to the variance of POI and non-POI data.