Enron Submission Free-Response Questions

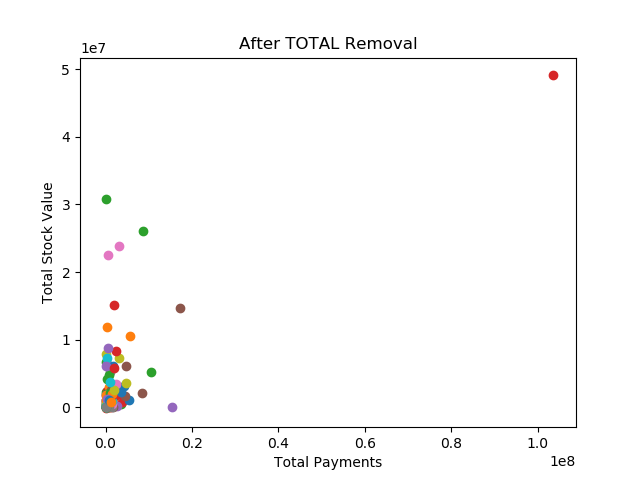
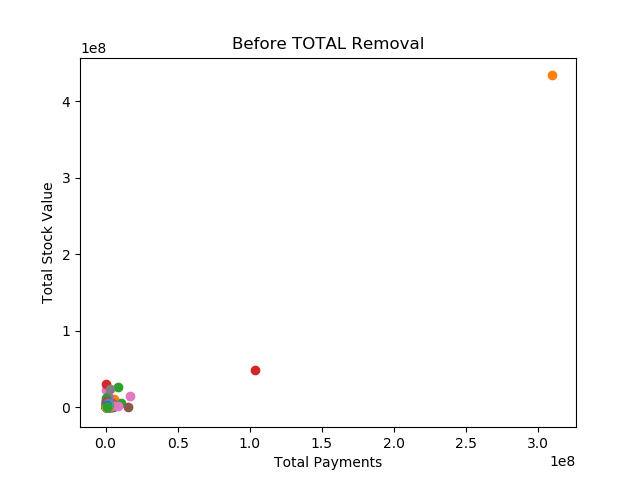
1. Summarize for us the goal of this project and how machine learning is useful in trying to accomplish it. As part of your answer, give some background on the dataset and how it can be used to answer the project question. Were there any outliers in the data when you got it, and how did you handle those?  [relevant rubric items: “data exploration”, “outlier investigation”]

*The goal of this project is to use the Enron dataset to determine persons of interest (POIs) using a machine learning algorithm. During the investigation into the company, the FERC created the Enron dataset. It contains roughly 146 users and those users are mostly senior management. As for the actual POIs, there were 18. I used 20 features of those given by Udacity.*

*The sheer size of this dataset is useful in applying machine learning algorithms as regular techniques would take too long and probably be prone to errors. Also, machine learning is valuable in researching this data if we don’t know exactly what we are looking for. We know we want to find the POIs but it is how we find them that makes machine learning application so valuable. It’s not as easy as following the money to the top executives, and in this case, we are following the emails.*

*There was an obvious outlier under salary called TOTAL. This field was the summary of all salaries and thus became a huge number that outweighed the other features. Once removed, we could see the data better and our algorithms worked better.*

*A scatter plot was a good way to visualize any outliers in the data. From there I went through each key to ignore NaN values and then sorted them. TOTAL was the highest value in the dictionary.*

**

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 used a combination of financial and email features (19 in total, dropping email\_adddress) which were given and created 2 more features – ratio\_from\_poi and ratio\_to\_poi. The Enron dataset is used mostly to find those people of interest who were essentially outliers in their own organization. Determining who sent emails to them (to\_messages) and from them (from\_messages) are major indicators to those exploring this data for insights. I figured the best way to do this was to create ratios of the from/to people to the POIs.*

*In order to further eliminate some of the weaker features, I scored the features using SelectKBest, using the classif function with a k = 19, and from there took the initial list of 19 features down to 9. The newly created ratio\_to\_poi became the 5th highest scored feature (Score: 16.641).*

1. *exercised\_stock\_options - 25.09754152873549*
2. *total\_stock\_value - 24.4676540475264*
3. *bonus - 21.06000170753657*
4. *salary - 18.575703268041785*
5. ***ratio\_to\_poi - 16.64170707046899***
6. *deferred\_income - 11.5955476597306*
7. *long\_term\_incentive - 10.072454529369441*
8. *restricted\_stock - 9.3467007910514881*
9. *total\_payments : 8.866721537107772*

*The features contained in this dataset mostly used different units and different values. For that reason, transforming them using sklearn.preprocessing MinMaxScaler was applied to the revised features.* *Feature scaling was also utilized as there were several outliers (beyond what was removed already) which could skew the results but due to the validity of the data (Jeffrey Skilling e.g.), these points could not be removed as they were key indicators of the data we are working with.*

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 chose the NaiveBayes algorithm as the accuracy was one of the highest, I tested at 85.7%. It also had a well round set of metrics with room for improvement during the tuning portion of this analysis. I also used Decision Tree, AdaBoost, Random Forest, and K-Nearest Neighbors algorithms. I tried many different parameters for each and went back and forth between the accuracy and f1 score to determine which algorithm to ultimately choose.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | *Accuracy* | *Precision* | *Recall* | *F1 Score* |
| *DecisionTree* | 0.6667 | 0.1666 | 0.3333 | 0.2222 |
| ***NaiveBayes*** | **0.8571** | **0.5** | **0.5** | **0.5** |
| *Random Forest* | 0.9047 | 1 | 0.3333 | 0.5 |
| *Adaboost* | 0.85714 | 0.5 | 0.3333 | 0.4 |
| *kNN* | 0.7857 | 0.2 | 0.1666 | 0.1818 |

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”]

*I tuned my Naïve Bayes algorithm with 10-fold Stratified Shuffle Split and GridSearchCV (using the var\_smoothing parameter). The final accuracy I got after tuning my model ranged between 85.5-88.7% which was approximately a 3% accuracy increase.*

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 taking a trained model and evaluating against a test dataset. A common mistake with validation for a model is to not split up the data into testing and training datasets. I opted to allocate 30% of the data to my test data.*

*I used sklearn.model\_selection’s cross validation to find the mean accuracy of 30% of our data which returned a much improved 88.7% accuracy to our dataset.*

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”]

*After tuning our Naïve Bayes classifier using our hyperparameters, we received a 2.3% boost in our accuracy using our feature set. Our precision score also increased to 91% from 50%, Recall decreased to 12% from 50% and the f1 score decreased from 50% to 21%.*

*Precision is the number of correct results divided by the number of all returned results. In our case, we were able to select 91% relevant persons of interest from the data set.*

*Recall is the measurement of how many relevant POIs were selected. After selecting 66% of relevant persons using our precision score, we could successfully identify 12% of them as true persons of interest.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Accuracy* | *Precision* | *Recall* | *F1* | *F2* |
| 0.88113 | 0.90943 | 0.1205 | 0.2128 | 0.1458 |
| *Total Predictions* | *True Positives* | *False Positives* | *False Negatives* | *True Negatives* |
| 15000 | 241 | 24 | 1759 | 12976 |