Machine Learning Project: Identify Fraud from Enron Email

Amandeep Mander

The goal of this project is to identify employees from Enron that have been in on the fraud committed that came to light in 2001. This will be based on an algorithm using public Enron financial and email information.

Part 1

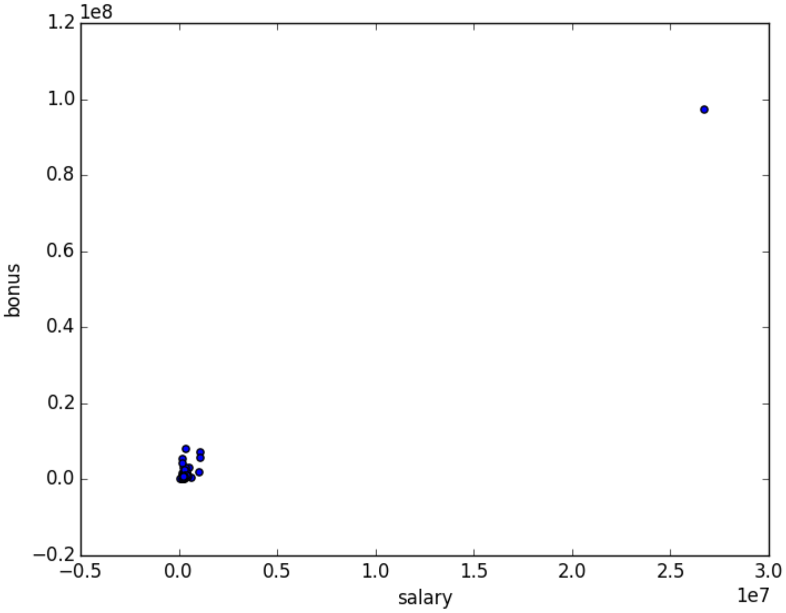
Data Exploration

The dataset that will be used in this investigation to identify employees that may have committed fraud within Enron will be publicly available financial and email information. By breaking up the dataset into training and test sets and using our set of known POIs (Person Of Interest), we can use the features associated with each employee to identify whether or not they may be part of the fraud. This dataset contains a total of 146 points. Out of the 146 observations, 18 (12.3%) are POIs and 128 (87.7%) are therefore non-POIs. Each of the observations contains 21 features. Since this is a real-world dataset, not all of the features have attached values to them, hence showing up as NaNs. The list below provides the top 5 features the most missing values:

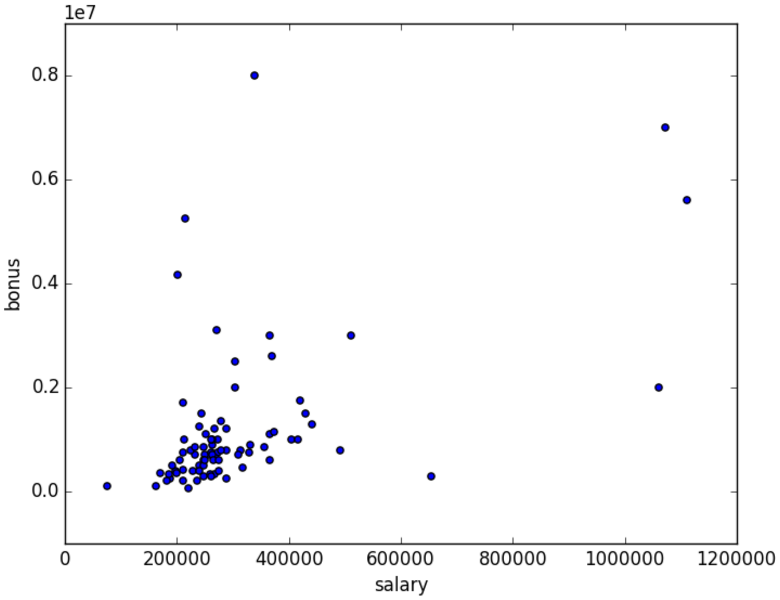
1. loan_advances: 142 NaNs
2. director_fees: 129 NaNs
3. restricted_stock_deferred: 128 NaNs
4. deferral_payments: 107 NaNs
5. deferred_income: 97 NaNs

Outlier Investigation

In order to scan for any potential outliers in the dataset, we first look at a scatterplot between the salary and bonus.



Cleary, there is an outlier in the top-right corner of the plot that requires further examination. This point happens to be the 'TOTAL' line in the financial information that sums up the various financial metrics within our dataset. Since this observation does not correspond to an employee, it should be removed from the data. There is another observation in the dataset called 'THE TRAVEL AGENCY IN THE PARK' that also should be removed. After removing these 2 observations, we duplicate the plot above and notice that there are no points that are extremely different than the rest.



Part 2

Create New Feature

In an attempt to increase the accuracy of the algorithm to identify the POIs in the dataset, a new feature was created. This feature is the ratio between the bonus and the salary of each employee. The rationale behind this is that if an employee has an unusually large bonus to salary ratio, they must rely a lot more on their bonus than their salary for a higher income and would therefore be more inclined to engage in illegal or unethical behavior to boost company performance, which would then in turn lead to a higher bonus.

Feature Selection

The features used in the algorithm were determined using the SelectKBest method. Below is a table that shows the accuracy, precision and recall scores for values of K between 1 and 10 achieved on the test set.

K	Accuracy	Precision	Recall
1	0.90	0.46	0.32
2	0.84	0.47	0.27
3	0.84	0.49	0.35
4	0.85	0.50	0.32
5	0.85	0.49	0.38
6	0.85	0.49	0.38
7	0.85	0.46	0.37
8	0.85	0.46	0.38
9	0.84	0.38	0.32
10	0.84	0.37	0.31

It can be seen that K = 5 and K = 6 have the highest accuracy, precision and recall scores. In order to avoid over fitting the model, it is better to go with the model with less features given equal performance. The table below outlines which features were selected for the final algorithm along with their scores.

Feature	Score
exercised_stock_options	1.0
total_stock_value	0.98
bonus	0.86
salary	0.78
deferred_income	0.53

The manually created bonus to salary ratio ended up being the 6th most important features with a score of 0.50. If this feature is not included in the final algorithm, it is replaced with long_term_incentive with a slightly lower score of 0.47 with the accuracy, precision and recall scores now becoming 0.85, 0.49 and 0.38, respectively.

Feature Scaling

Applying a Min-Max Scaler to the algorithm produces the following accuracy, precision and recall scores:

Accuracy	**Precision**	**Recall**
0.85	0.49	0.38

It turns out that the accuracy, precision and recall scores remain the same with the deployment of feature scaling. This means that featuring scaling did not add to better performance of our model. Since the purpose of feature scaling is to normalize the feature values, it turns out in our case that the large range of the values plays a crucial role in separating the POIs from the non-POIs. Therefore, feature scaling was not used in the final algorithm.

Algorithm Selection

The algorithm that I ended up using to classify POIs and non-POIs was the Gaussian Naïve Bayes algorithm. The other algorithm that I tried was the Decision Tree Classifier. The performance of each algorithm is summarized below:

Algorithm	Accuracy	Precision	Recall
Naïve Bayes	0.85	0.49	0.38
Decision Tree	0.85	0.43	0.13

Parameter Tuning

Tuning the parameters of an algorithm simply means going through the process of optimizing the parameters of your algorithm that provides the best performance. In our case, this means selecting the parameters of our algorithm that maximize the accuracy, precision and recall scores in the classification of POIs and non-POIs. If the parameters are not tuned correctly, the algorithm will not perform optimally and thus have a higher likelihood of misclassifying the employees. I used parameter tuning for the Decision Tree Classifier in order to ensure the most accurate model was utilized. Below is a list of the parameters that were used tuned, along with the range of possible values:

Parameter	Possible Values
tree__criterion	gini, entropy
tree__splitter	best, random
tree__min_samples_split	2, 10, 20
tree__max_depth	10, 15, 20, 25, 30
tree__max_leaf_nodes	5, 10, 30

In order to optimally tune the parameters for my Decision Tree Classifier, I used the GridSearchCV function.

Validation

Validation is the process in which we use a subset of our training data to tune the parameters of the algorithm. If this step is done incorrectly, you run the risk of over fitting your model that will perform very good on the training set but will do very poorly on the test set. For my algorithm, I used the StratifiedShuffleSplit function to split the data into training and tests for validation purposes.

Evaluation

I evaluated by algorithm based on the precision and recall scores. For my algorithm, I was able to achieve precision and recall scores of 0.49 and 0.38, respectively. Precision refers to the probability that an employee identified as a POI is actually a POI and recall refers to the probability of my algorithm in positively identifying a POI.