PROJECT REPORT- Intro To Machine Learning

Rajiv Kumar

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”]

Answer:

The goal of the project is deploy a supervised machine learning algorithm to identify POI(Person of interest) from the enron dataset. POI is the person who was potentially involved in the enron fraud case. Following is the summary of the Enron Data set

- Data set contains total 146 rows and 22 attributes. Following is the list of attributes.

'Name', 'salary', 'to_messages', 'deferral_payments', 'total_payments', 'exercised_stock_options', 'bonus', 'restricted_stock', 'shared_receipt_with_poi', 'restricted_stock_deferred', 'total_stock_value', 'expenses', 'loan_advances', 'from_messages', 'other', 'from_this_person_to_poi', 'poi', 'director_fees', 'deferred_income', 'long_term_incentive', 'email_address', 'from_poi_to_this_person'

Enron dataset contain some key characterstics , which can be used to identify POI. The POI attribute , which is already defined, identifies the important characteristics of POI and we can use machine learning algorithm to find patterns among the POIs and train our model to predict whether a person is POI or not based on given characterstics.For Example, Long_term_incentives for POIs seems to be higher than other employees so this could be considered as one of the attribute to identify what other employees have similar characterstics.

Data Exploration:

First Step:

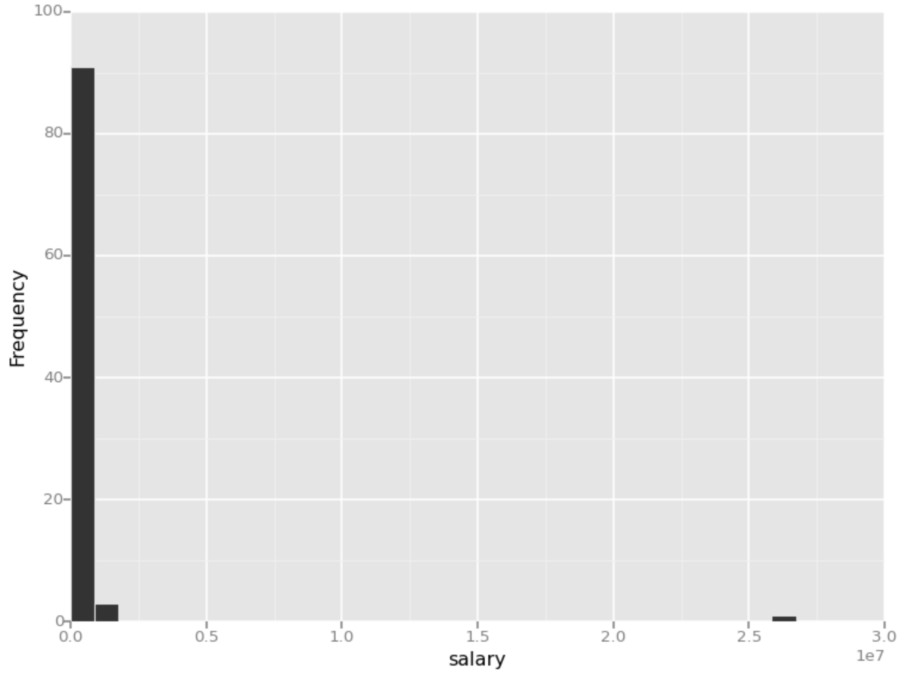
In my explanatory analysis , I started with the descriptive statistics for all attributes in the enron data set. Here are my key findings:

- Max value for each attribute was exceptionally high(way higher than the interquartile Range), indicating the presence of outliers
- Count of non missing values was significantly lower than total rows in the data set, which is an indication that there are lot of missing values in the dataset.
- I briefly looked at the median and mean values for the attributes and found that the median value for all the attributes was significantly less than the mean value. For example: Median Value for the salary column was only 259996, however the mean value was 562194. This was a strong indication that data is positively skewed and there are certainly outliers in the data.

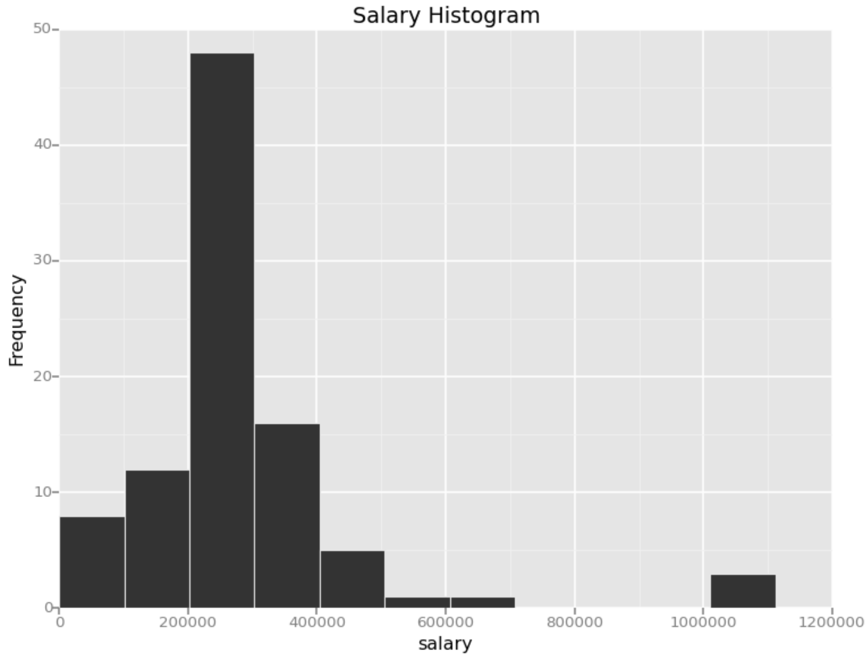
Second Step:

To further investigate the data, I plotted some graphs. Below are my key findings;

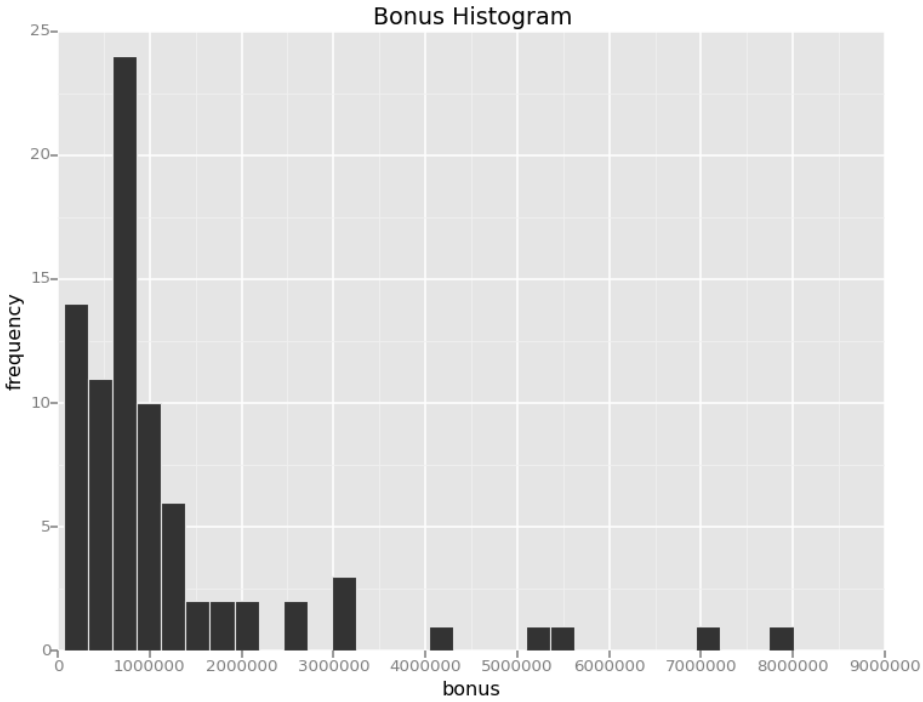
I started by drawing the histograms of the key attributes to see how the data is distributed. Below is one such key graph to show the trend In the data.

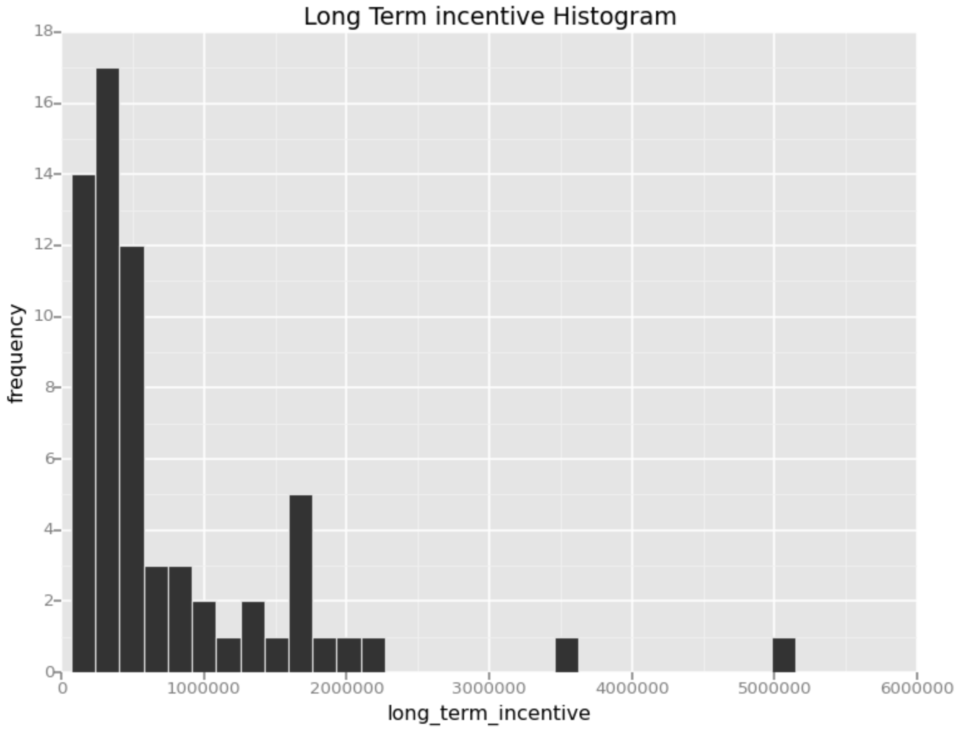


This graph indicated an outlier in the salary attribute. In my further investigation , I found that there is a row “Total” in the data set , which contains the total for each attribute . I removed the “Total” row from the data set and my new histogram for salary looked like following:



Even though the data is still positively skewed, I decided not to remove the outliers because these outliers are indeed key characteristics of POI. I noticed that same distribution of data in most of the attributes. Below are some of the other interesting histograms that helped in my analysis:





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”]

<u>Answer:</u>

In order to identify the key features , I continued with my exploratory data analysis and drew graphs of potential attributes with POI. I used Box plots to see that trend in the data for non poi vs poi. Below are some of the key graphs that helped me find some relationship between POI and the given attribute.

This graph clearly shows median salary is higher for POIs and is lower for non POIs. It also indicate the higher quartile range of salaries for POI then Non POIs. I drew other box plots to see the distribution of other key attributes with POI. Below are some key box plots that helped me in identifying the trend.

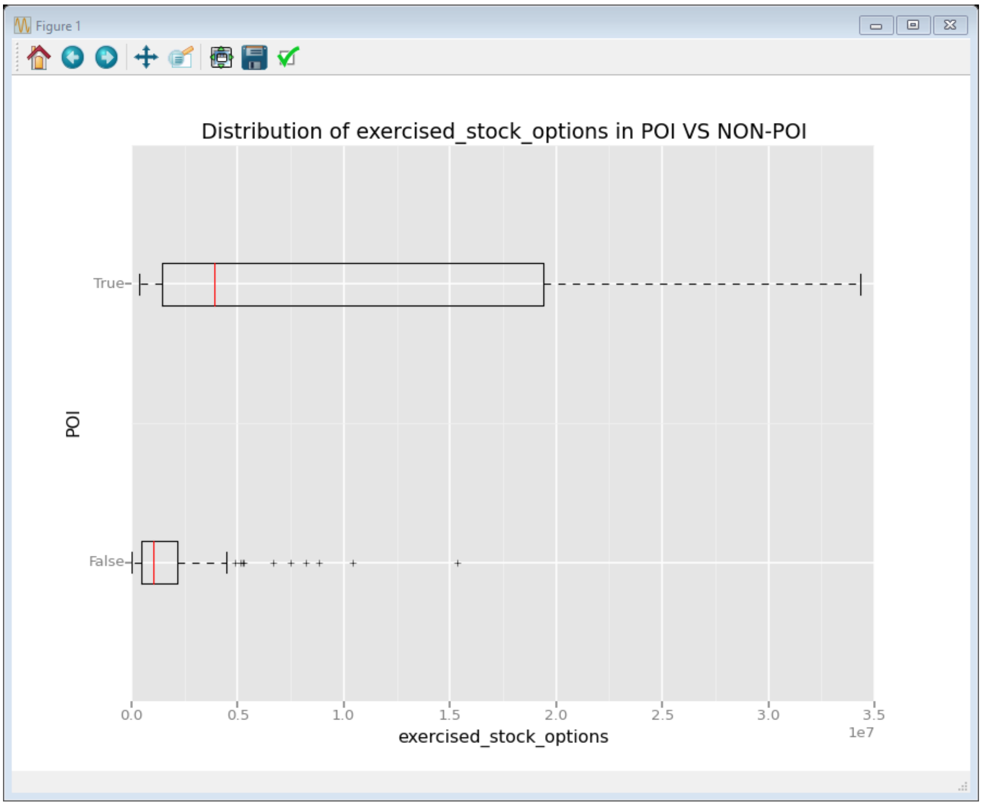


Figure 1
Distribution of exercised_stock_options in POI VS NON-POI
True
False
POI
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
1e7
exercised_stock_options

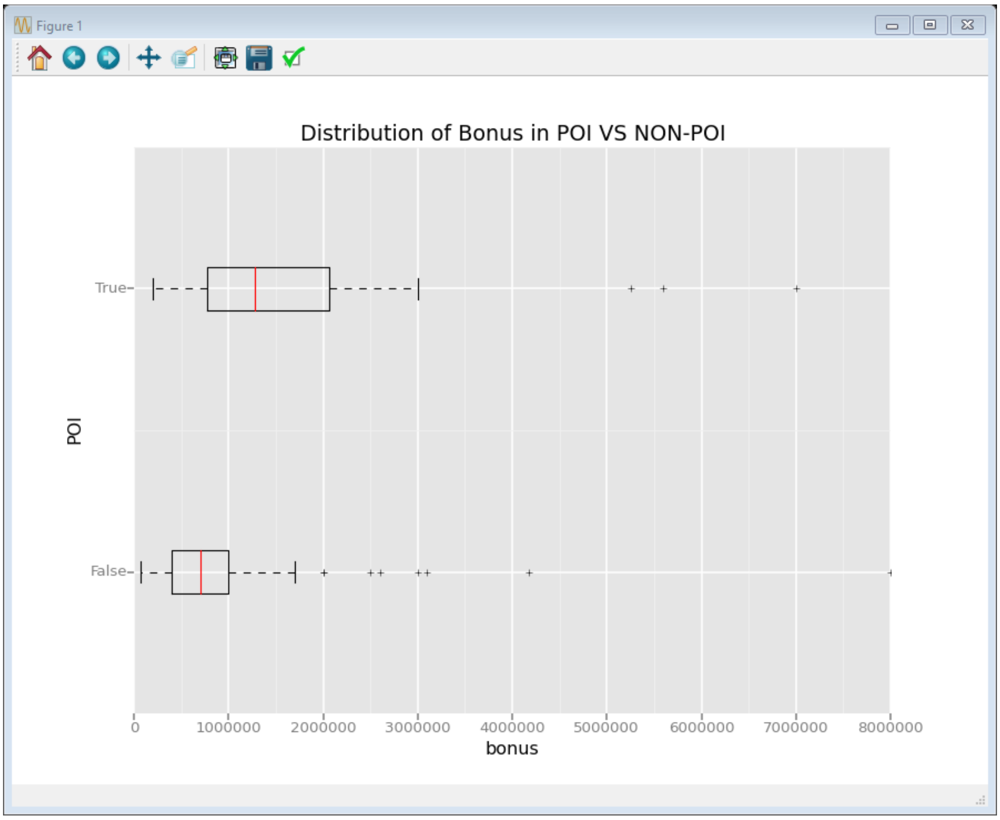
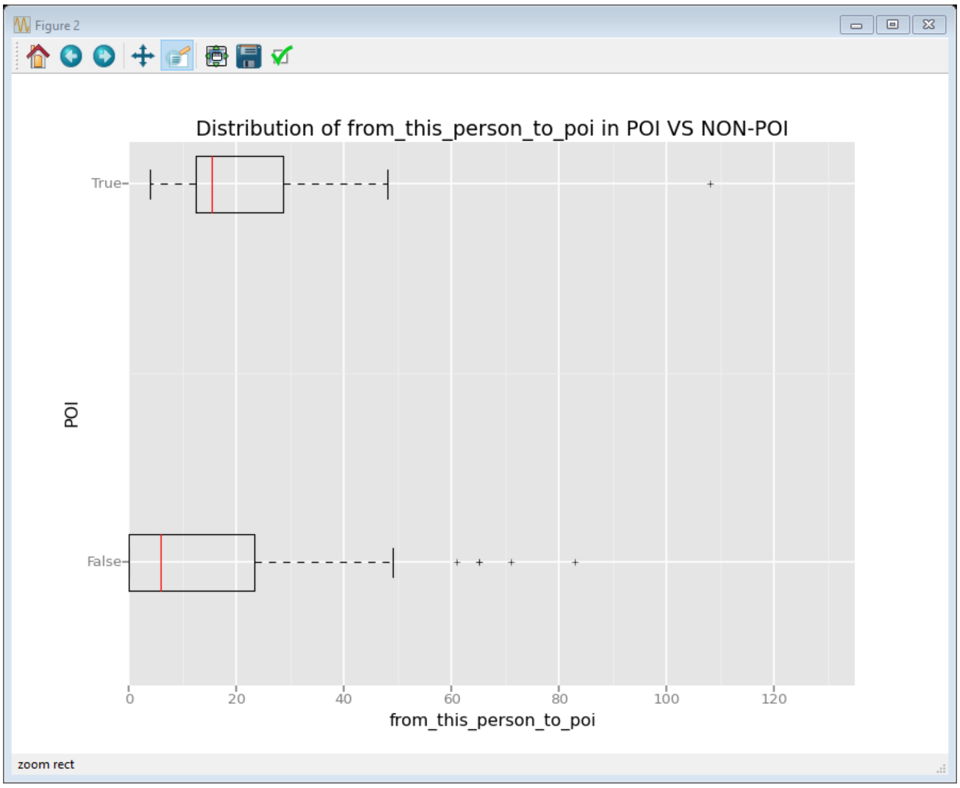


Figure 1
Distribution of Bonus in POI VS NON-POI
True
False
POI
0
1000000
2000000
3000000
4000000
5000000
6000000
7000000
8000000
bonus



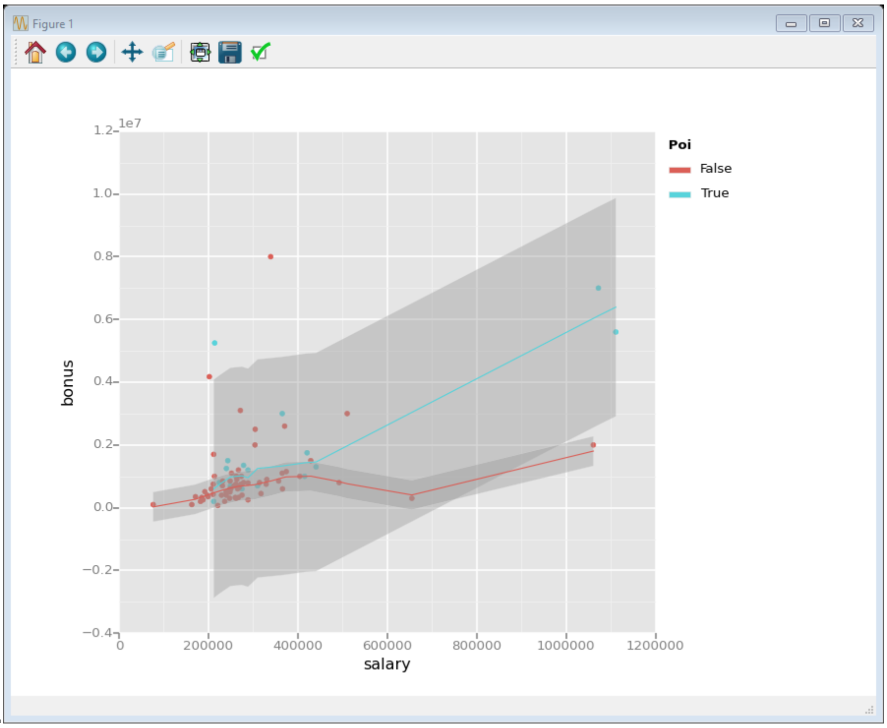
By my analysis so far, I knew that following attributes certainly play some important role in identifying a POI.

'poi','salary','bonus','long_term_incentive','exercised_stock_options','from_poi_to_this_person'.

I went ahead to look out for some interesting relationships among the key attributes . I started by using a correlation matrix on my data set. I found some interesting correlation between following attributes:

Salary , exercised_stock_options, bonus, total_stock_value , long_term_incentive and shared_receipt_with_poi , from_this_person_to_poi.

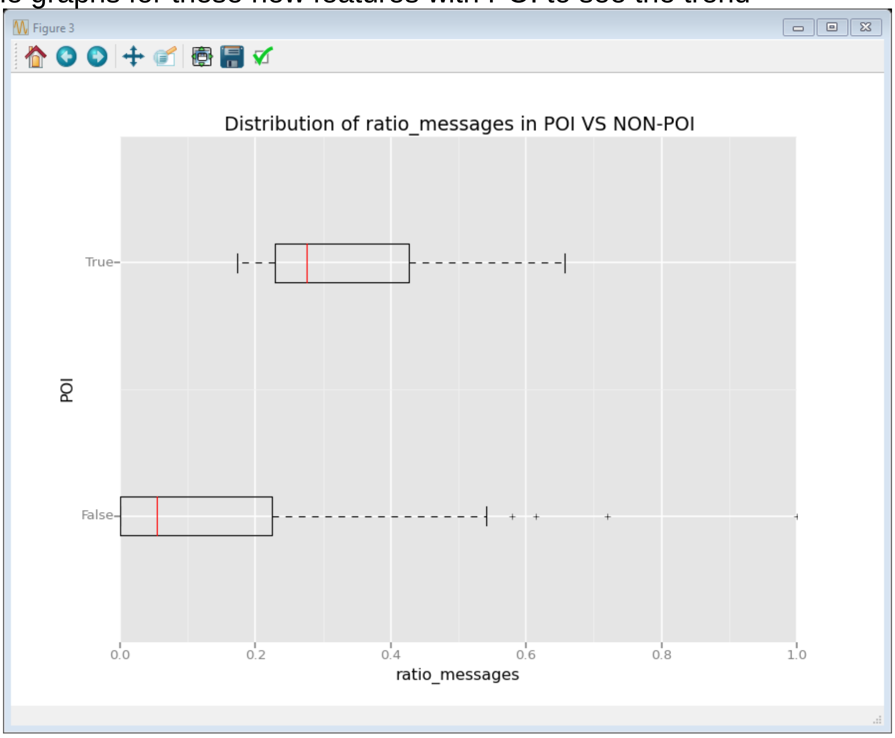
I did some multivariate analysis to see the underlying relationship. For example:Below graph helped me in visualizing a clear trend between salary and bonus for poi and non-poi.

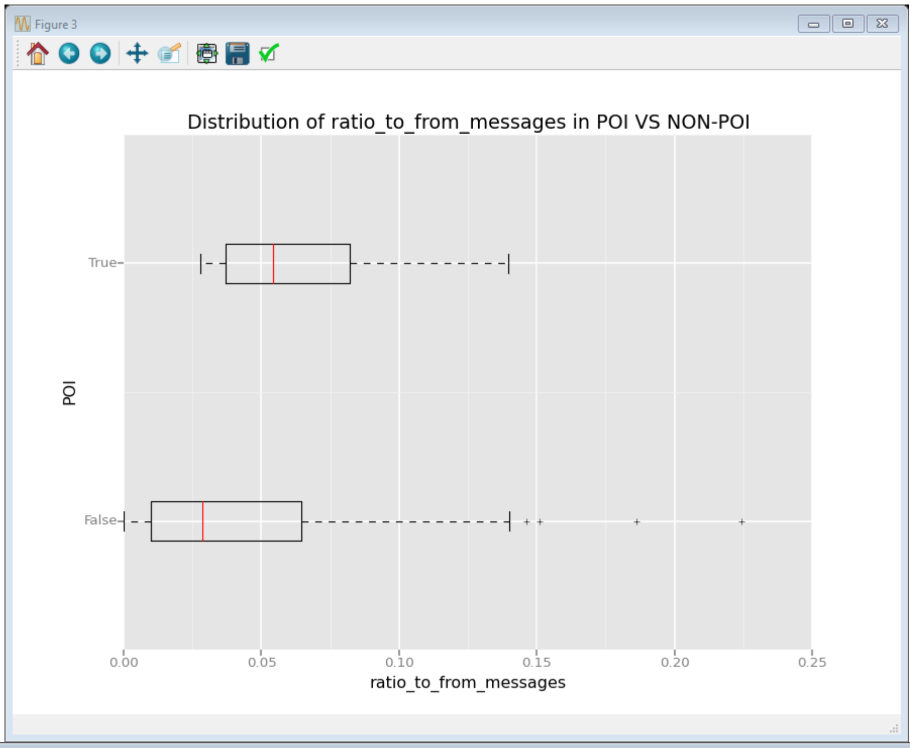


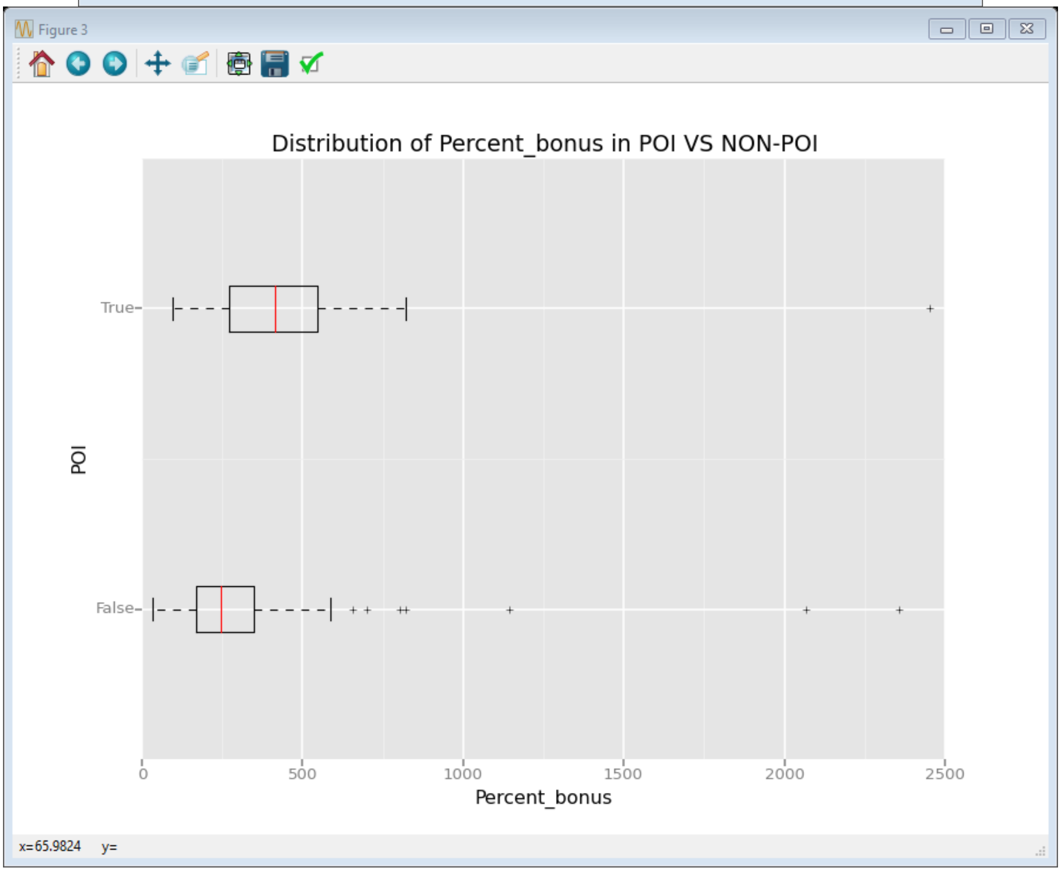
I also tried to engineer some new features based on current features. I develop following new featutres ;

- Percent_bonus=('bonus'/'salary')*100
- Ratio_message = from_this_person_to_poi / from_messages
- ratio_to_from_messages= (from_poi_to_this_person + from_this_person_to_poi)/ to_messages+from_messages

I draw the graphs for these new features with POI to see the trend







These graphs showed a clear trend and helped me in strengthening my intuition about key attributes.

I tried various combination of attributes to select the final list for my model. Various combinations of the features are provided in the further sections. Furthermore, I did not do any scaling of the features as it was not required for my algorithm choice. I used Guassian Naïve Bayes and Decision Tree classifier algorithms. Both these algorithm are not based on calculating the distance with the decision boundary therefore there was not going to be an impact of features magnitudes.

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"]

Answer:

After having an idea about my key attributes, I went ahead and tried couple of supervised machine learning algorithms. I also focused on tuning my algorithm by trying various combinations of key attributes. Here is my analysis:

I started by using Guassian naïve bayes algorithm ,as this algorithm is well suited for a supervised classification problem like this.

Below is the summary of my different attempts to choose the right algorithm and attributes:

Algorithm-GussianNB()	
Features	**Result**
'poi','salary','bonus','long_term_incentive','exercised_stock_options','ratio_messages'	Accuracy: 0.80050 Precision: 0.64985 Recall: 0.43800 F1: 0.52330 F2: 0.46855 Total predictions: 4000 True positives: 438 False positives: 236 False negatives: 562 True negatives: 2764
'poi','salary','bonus','long_term_incentive','exercised_stock_options','from_poi_to_this_person'	Accuracy: 0.81075 Precision: 0.69134 Recall: 0.43900 F1: 0.53700 F2: 0.47357 Total predictions: 4000 True positives: 439 False positives: 196 False negatives: 561 True negatives: 2804
'poi','salary','bonus','long_term_incentive','exercised_stock_options','ratio_to_from_messages'	GaussianNB() Accuracy: 0.78050 Precision: 0.58069 Recall: 0.43900 F1: 0.50000 F2: 0.46152 Total predictions: 4000 True positives: 439 False positives: 317 False negatives: 561 True negatives: 2683
'poi','salary','long_term_incentive','exercised_stock_options','ratio_to	Accuracy: 0.78075 Precision: 0.58554

_from_messages'	Recall: 0.42100 F1: 0.48982 F2: 0.44607 Total predictions: 4000 True positives: 421 False positives: 298 False negatives: 579 True negatives: 2702
'poi','salary','long_term_incentive','total_stock_value','ratio_to_from_messages'	Accuracy: 0.80660 Precision: 0.52973 Recall: 0.29400 F1: 0.37814 F2: 0.32272 Total predictions: 5000 True positives: 294 False positives: 261 False negatives: 706 True negatives: 3739
'poi','salary','Percent_bonus','long_term_incentive','exercised_stock_options','from_poi_to_this_person'	Accuracy: 0.81050 Precision: 0.68615 Recall: 0.44600 F1: 0.54061 F2: 0.47957 Total predictions: 4000 True positives: 446 False positives: 204 False negatives: 554 True negatives: 2796
'poi','bonus','long_term_incentive','exercised_stock_options','from_poi_to_this_person'	Accuracy: 0.83025 Precision: 0.77816 Recall: 0.44900 F1: 0.56944 F2: 0.49050 Total predictions: 4000 True positives: 449 False positives: 128 False negatives: 551 True negatives: 2872
['poi','Percent_bonus','long_term_incentive','exercised_stock_options','from_poi_to_this_person']	Accuracy: 0.85675 **Precision: 0.79367** Recall: 0.57700 F1: 0.66821 F2: 0.61032 Total predictions: 4000 True positives: 577 False positives: 150 False negatives: 423 True negatives: 2850
'poi','Percent_bonus','long_term_incentive','exercised_stock_options','ratio_messages'	Accuracy: 0.82275 Precision: 0.74872 Recall: 0.43800 F1: 0.55268 F2: 0.47764 Total predictions: 4000 True positives: 438 False positives: 147 False negatives: 562 True negatives: 2853

Algorithm-DecisionTreeClassifier	
Features	**Result**
'poi','salary','Percent_bonus','long_term_incentive','exercised_stock_options','from_poi_to_this_person'	Accuracy: 0.71875 Precision: 0.44175 Recall: 0.47400 F1: 0.45731 F2: 0.46718 Total predictions: 4000 True positives: 474 False positives: 599 False negatives: 526 True negatives: 2401
'poi','salary','bonus','long_term_incentive','exercised_stock_options','ratio_messages'	Accuracy: 0.66950 Precision: 0.33868 Recall: 0.33800 F1: 0.33834 F2: 0.33814 Total predictions: 4000 True positives: 338 False positives: 660 False negatives: 662 True negatives: 2340
'poi','salary','bonus','long_term_incentive','exercised_stock_options','from_poi_to_this_person'	Accuracy: 0.72950 Precision: 0.46139 Recall: 0.49000 F1: 0.47527 F2: 0.48400 Total predictions: 4000 True positives: 490 False positives: 572 False negatives: 510 True negatives: 2428
'poi','salary','long_term_incentive','exercised_stock_options','ratio_to_from_messages'	Accuracy: 0.73075 Precision: 0.45528 Recall: 0.39200 F1: 0.42128 F2: 0.40321 Total predictions: 4000 True positives: 392 False positives: 469 False negatives: 608 True negatives: 2531
'poi','salary','long_term_incentive','exercised_stock_options','ratio_to_from_messages'	Accuracy: 0.72950 Precision: 0.45244 Recall: 0.39000 F1: 0.41890 F2: 0.40107 Total predictions: 4000 True positives: 390 False positives: 472 False negatives: 610 True negatives: 2528
['poi','salary','long_term_incentive','total_stock_value','ratio_to_from_messages']	Accuracy: 0.65800 Precision: 0.22901 Recall: 0.30000 F1: 0.25974 F2: 0.28249 Total predictions: 5000

	True positives: 300 False positives: 1010 False negatives: 700 True negatives: 2990
'poi','Percent_bonus','long_term_incentive','exercised_stock_options','ratio_messages'	Accuracy: 0.68800 Precision: 0.38988 Recall: 0.43900 F1: 0.41298 F2: 0.42821 Total predictions: 4000 True positives: 439 False positives: 687 False negatives: 561 True negatives: 2313
'poi','Percent_bonus','long_term_incentive','exercised_stock_options','from_poi_to_this_person'	Accuracy: 0.72975 Precision: 0.46211 Recall: 0.49400 F1: 0.47753 F2: 0.48728 Total predictions: 4000 True positives: 494 False positives: 575 False negatives: 506 True negatives: 2425
'poi','bonus','long_term_incentive','exercised_stock_options','from_poi_to_this_person'	Accuracy: 0.74325 Precision: 0.48730 Recall: 0.51800 F1: 0.50218 F2: 0.51155 Total predictions: 4000 True positives: 518 False positives: 545 False negatives: 482 True negatives: 2455

I ended up using GuassianNB() algorithm as this algorithm gave me higher precision with given combination of my key attributes.

['poi','Percent_bonus','long_term_incentive','exercised_stock_options','from_poi_to_this_person']	Accuracy: 0.85675 **Precision: 0.79367** Recall: 0.57700 **F1: 0.66821** F2: 0.61032 Total predictions: 4000 True positives: 577 False positives: 150 False negatives: 423 True negatives: 2850

As shown in above table , I also tried using DecisionTreeClasifier but could not get good precision. Performance difference between these two algorithms is shown in above table.

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"]

Answer:

Many Machine learning algorithms are parameterised and modification of the parameters can influence the outcome of the learning process. Having one or two algorithm that perform well can be a good start but sometimes parameter tuning can yield best results from these selected algorithms. Each parameter in the algorithm can be considered as a dimension on the graph with values of the parameters as a point along the axis. So n parameters in an algorithm can be considered as n dimensional cube of possible configurations. The objective of tuning the algorithm is to find the best point in the n dimensional cube for the given problem. If we don't tune our algorithm well then we can miss the opportunity of achieve the best performance from our model.

I ended up choosing GuassinNB() as it gave me better performance but I tried tuning my decision tree classifier . I used GridSearchCV to get the best combinations of the parameters. I passed following list of parameters to get the best combination.

parameters = {'criterion':('gini', 'entropy'),'splitter':('best','random'), 'min_samples_split':[2,3,4,5,6,7,8,9,10]}

Below is the comparison of the results that I got using the algorithm with and without tuning. Clearly ,Tuning the algorithm helped in better performance as we can see.However, F1 Score fell a little but after tuning but precision has gone up, which is of our interest in this particular scenario.

Algorithm-Decision Tree Classifier		
Features	**Without Tuning**	**With Tuning**
`'poi','bonus','long_term_incentive','exercised_stock_options','from_poi_to_this_person'`	Accuracy: 0.74325 Precision: 0.48730 Recall: 0.51800 F1: 0.50218 F2: 0.51155 Total predictions: 4000 True positives: 518 False positives: 545 False negatives: 482 True negatives: 2455	Accuracy: 0.76325 Precision: 0.53557 Recall: 0.39900 F1: 0.45731 F2: 0.42044 Total predictions: 4000 True positives: 399 False positives: 346 False negatives: 601 True negatives: 2654

In some cases, Tuning the algorithm can help improve the performancy drastically. Given the number of parameters, sometime it is obvious to tune couple of parameters manually and see the impact. In other situations, algorithms such as GridSearchCV can be used to tune the algorithm for better performance.

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"]

   Answer:

Validation is a process of gauging the performance of your model against a test data set. A test data is set of data points that the model has not seen before. The classic mistake during validation step is to validate the model on the training data set. Under such situation , we can end having a over fitted model, which perform exceptionally well on the training data set but does not perform well on the test data set.

In the given problem, I used the tester.py script to validate the results of my model. Given the low number of data points, the strategy was to use stratified sampling technique for validation i.e K-Fold Validation technique with 1000 folds.

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"]

Answer:

Following are some useful evaluation matrices.

Accuracy : Accuracy for a given algorithm helps in identifying how many times, the algorithm was able to make correct predictions out of given total of tries. It is a ratio of correct predictions to total tries.However , in some situation where number of data points are less , Accuracy may not be a good measure to gauge the performance of the algorithm. For example, lets say we have following prediction and test_labels

Predictions	0	1	1	1	0
Test_lables	0	0	0	1	1

Here we have only 5 data points and accuracy is ½=50% even though the model was able to predict the correct value only once. So this situation can lead to a false impression that model is doing well.

Due to the reason mentioned above, I used precision and recall to measure the performance of the algorithm in the given problem.

Precision:

Precision is also called positive predictive value of an algorithm. It is ration of number of instances that algorithm has predicted correctly and Total number of positively reported cases by the algorithm. i.e True Positive/ (True Positive + False Positive). I have focussed on achieving a high precision that means that whenever a POI gets flagged in my test set, I know with a lot of confidence that its very likely to be a real POI and not a false alarm. In the given problem , we have precision value of 0**.79** ,which means that we can say that ~80 % of the time POI predicted by algorithm is actually a true POI.

['poi','Percent_bonus','long_term_incentive','exercised_stock_options','from_poi_to_this_person']	Accuracy: 0.85675 **Precision: 0.79367** **Recall**: 0.57700 **F1: 0.66821** F2: 0.61032 Total predictions: 4000 True positives: 577 False positives: 150 False negatives: 423 True negatives: 2850

Recall: Recall also known as sensitivity of an algorithm is the ratio of how many times algorithm has predicted a correct values and total number of correct values predicted + number of time it has missed to predict correct values i.e True Positive / True Positive + False Negatives

That means that, nearly every time a POI shows up in my test set, I am able to identify him or her. The cost of this is that I sometimes get some false positives, where non-POIs get flagged. In the give problem, we have achieved a Recall value 0.57, which means that algorithm will be able to find the POI at least 57% times.

['poi','Percent_bonus','long_term_incentive','exercised_stock_options','from_poi_to_this_person']	Accuracy: 0.85675 **Precision: 0.79367** **Recall**: 0.**57700** **F1: 0.66821** F2: 0.61032 Total predictions: 4000 True positives: 577 False positives: 150 False negatives: 423 True negatives: 2850