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 characteristics.

Below is Brief summary of the data:

* total number of data points- Total 146 Records and 22 attributes. Out of these there are total 18 POIs
* allocation across classes (POI/non-POI) :18 POIs and 128 NON POIs i.e only 12% Records are POIs and rest ~88% are NON POIs
* number of features used : There are total 22 features in the data set.
* are there features with many missing values? etc. :There are lot of missing values in the columns. For exa: loan\_advances , director fees and restricted\_stock\_defferred columns has mostly missing values.

Data Exploration:

**First Step:**

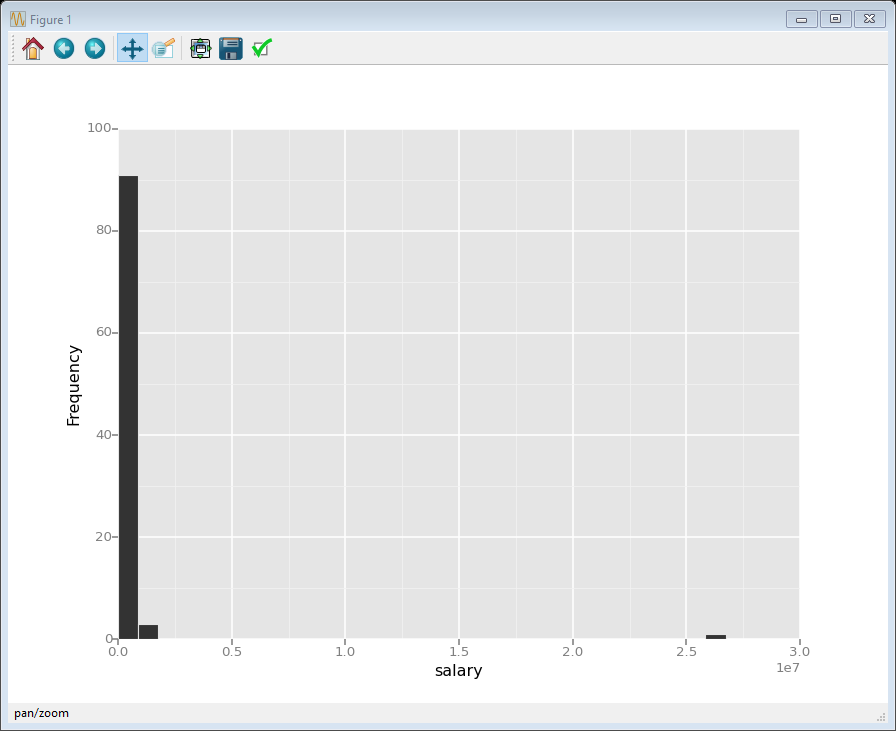
In my explanatory analysis , I started with the descriptive statistics for all attributes in the enron data set. Here are my key findings:

* 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.

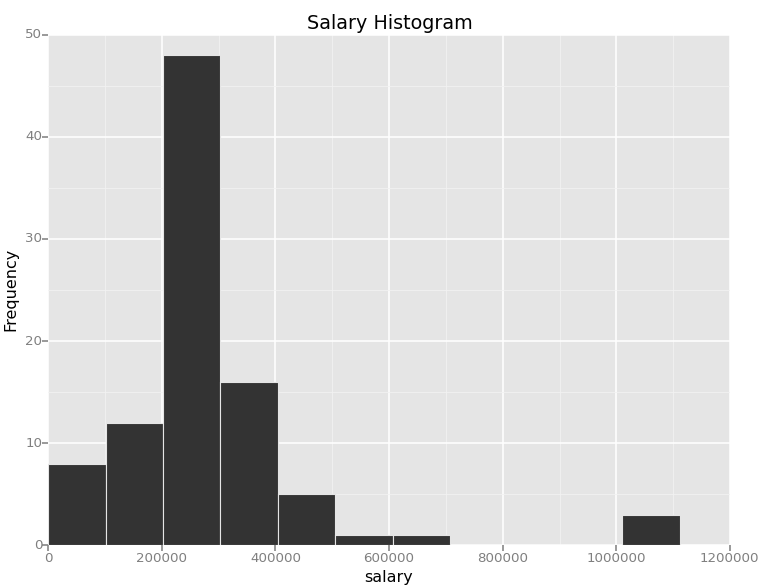
**Second Step:**

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

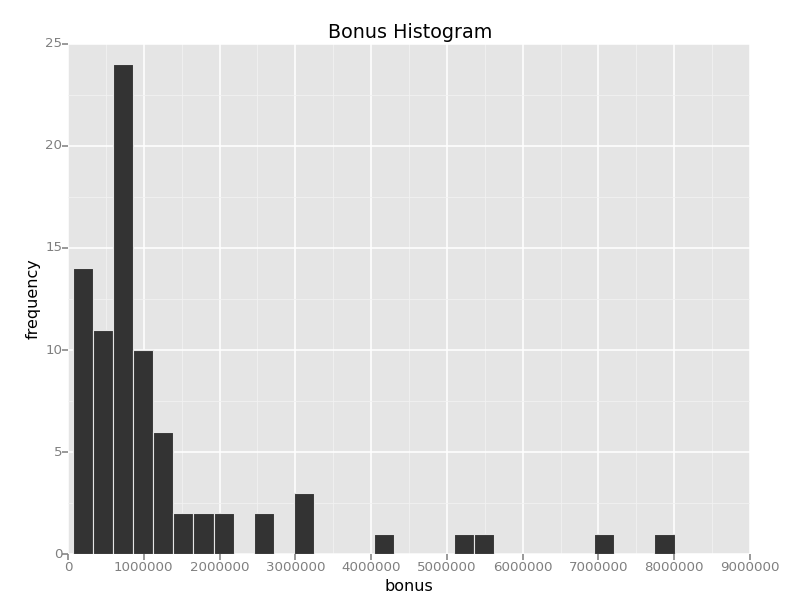
I started by drawing the histograms of the key attributes to see how the data is distributed. Below is one such key graph to show the trend In the data.

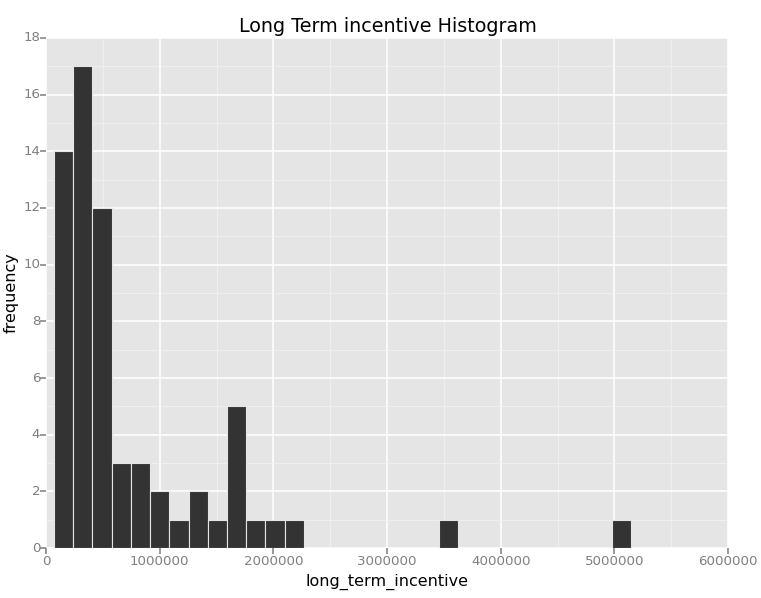


This graph indicated an outlier in the salary attribute. In my further investigation , I found that there is a row “Total” in the data set , which contains the total for each attribute . I removed the “Total” row from the data set and my new histogram for salary looked like following:



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:



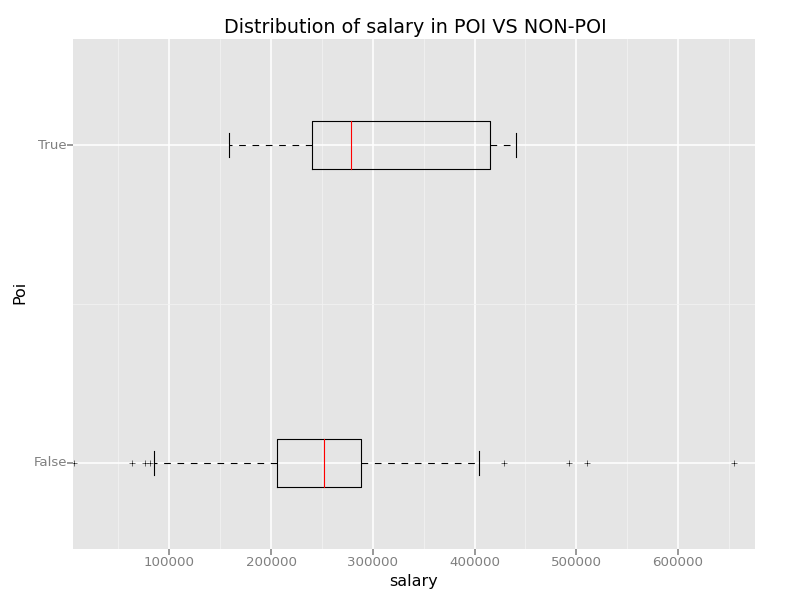


Also, there are lot of missing values present in the columns , in order to successfully deploy the machine learning algorithm I have decided to remove the missing values from the data set. We could have tried filling the missing values with the mean or median of a the given column but since the data is so much skewed, it would not be appropriate. Therefore, I decided to drop the values even though it is going to reduce the overall size of my data set drastically.

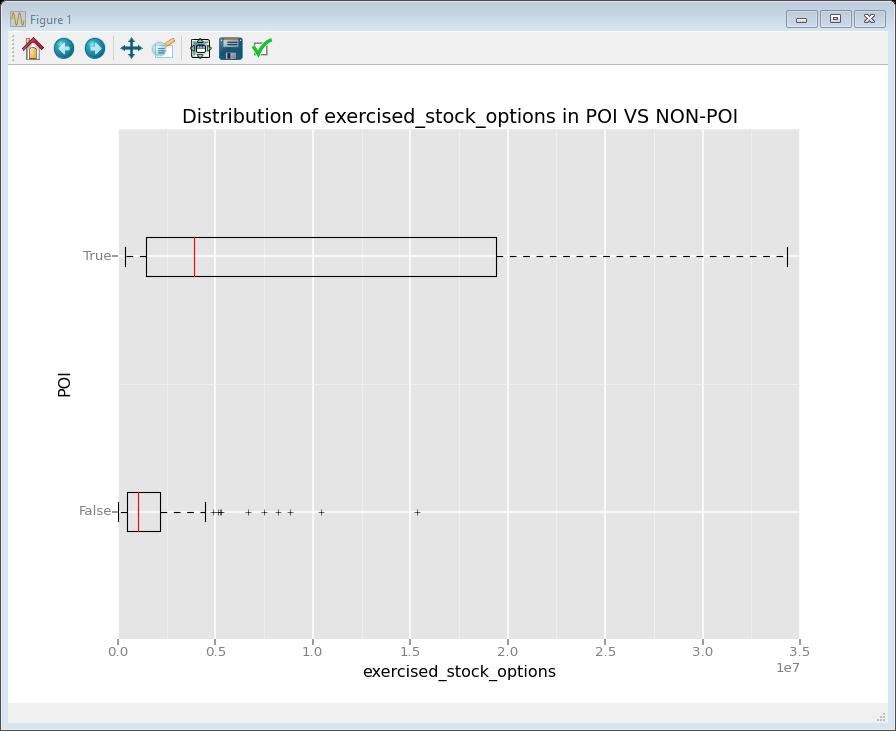
1. What features did you end up using in your POI identifier, and what selection process did you use to pick them? Did you have to do any scaling? Why or why not? As part of the assignment, you should attempt to engineer your own feature that does not come ready-made in the dataset -- explain what feature 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”, “properly scale features”, “intelligently select feature”]

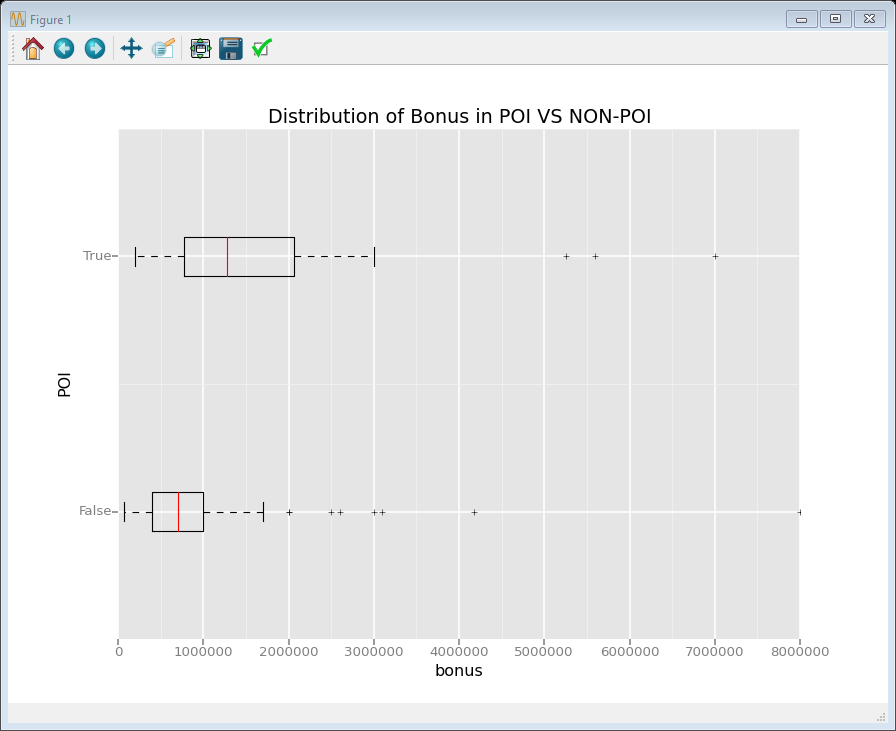
**Answer:**

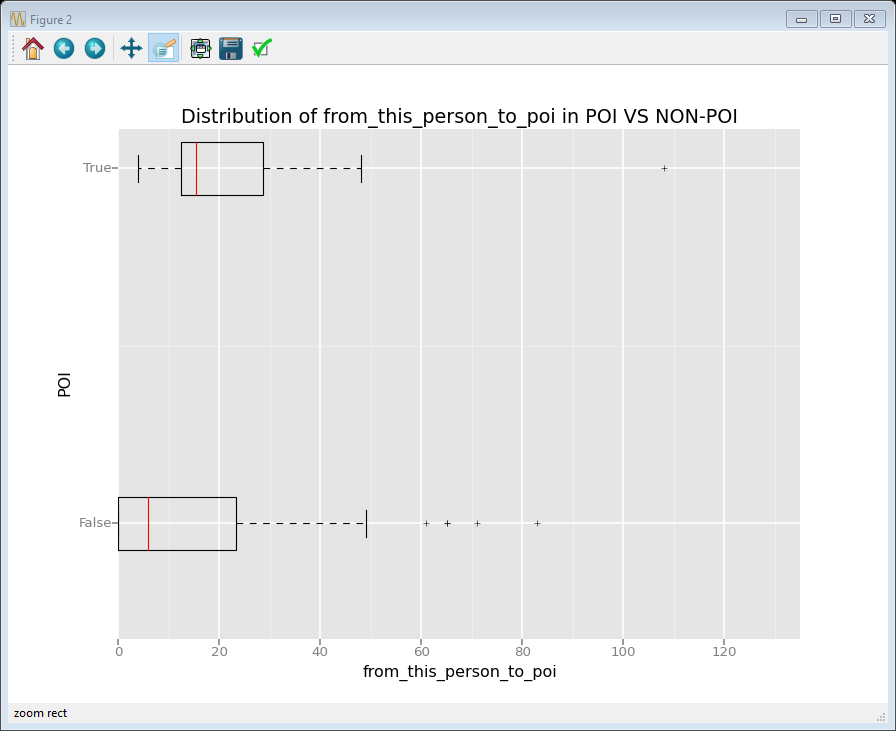
In order to identify the key features , I continued with my exploratory data analysis and drew graphs of potential attributes with POI. I used Box plots to see that trend in the data for non poi vs poi. Below are some of the key graphs that helped me find some relationship between POI and the given attribute.



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.







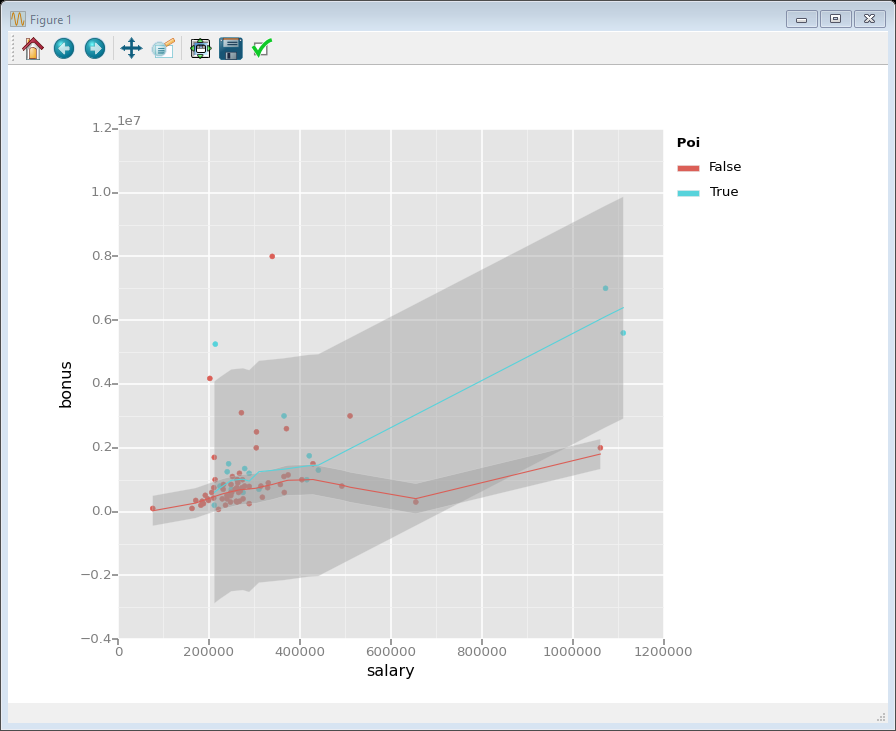
By my analysis so far, I knew that following attributes certainly play some important role in identifying a POI.

'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.

.

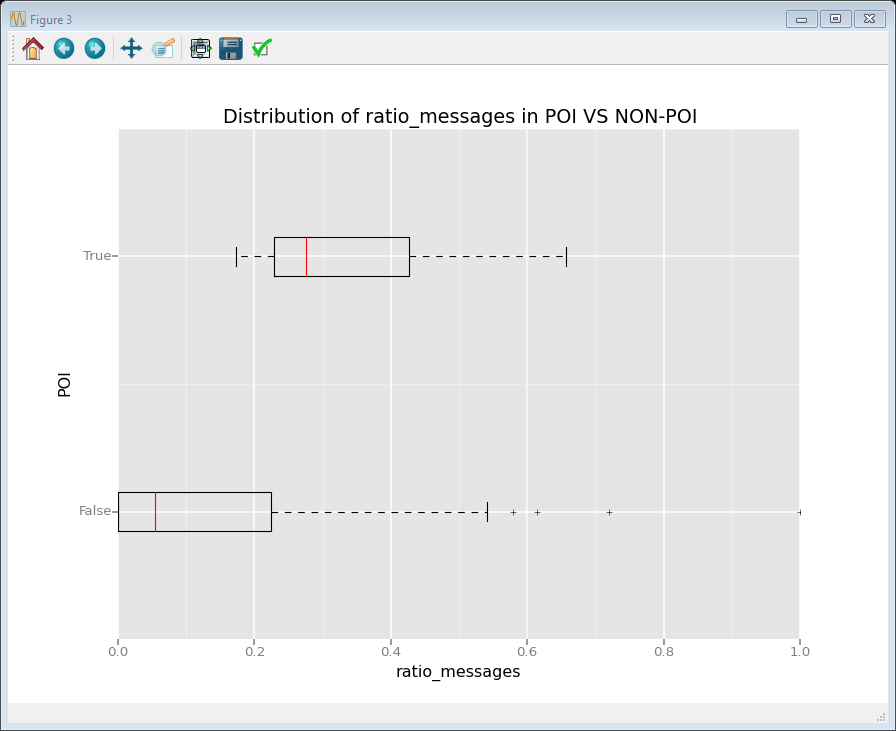
Based on my analysis so far , I observe that bonus and salary are correlated , which is intuitive as well. I tried to merge these two features in to one so that we don’t have redundant features in the model. I created Percent\_bonus feature, which reflect what % of bonus was given to the employee as shown below.

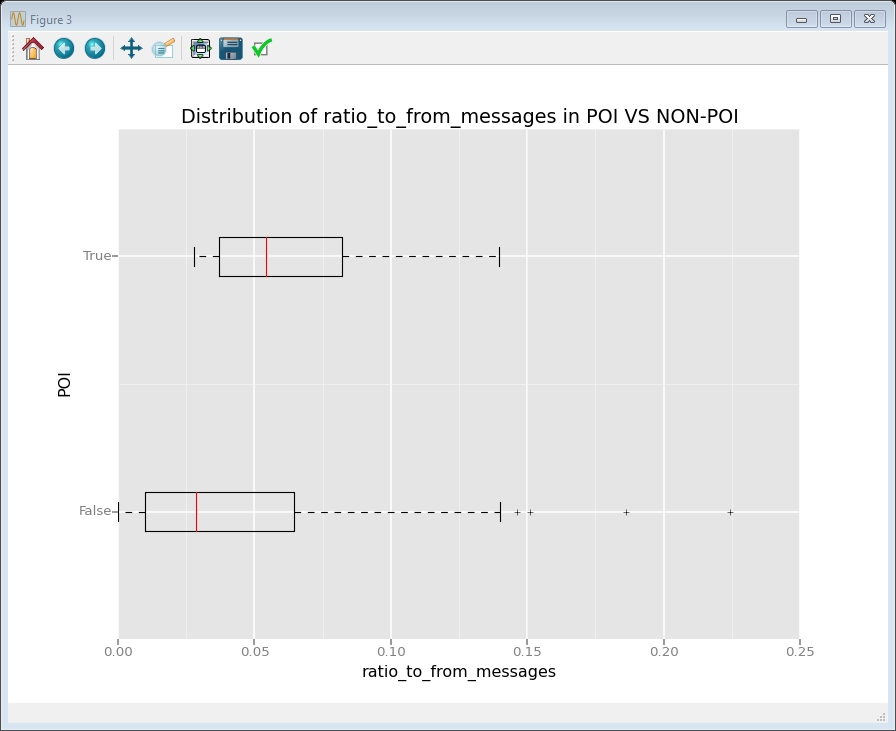
Also, I think it makes more sense to have the ratio of number of emails sent to POI vs total email sent by that person to have accurate numbers. For example: If the person has sent 1000 emails to the POI but in total has sent over 1 million emails then this person might not be the suspect.Hence I decided to create other two features , which are ratio\_message and ratio\_to\_from\_messages as shown below:

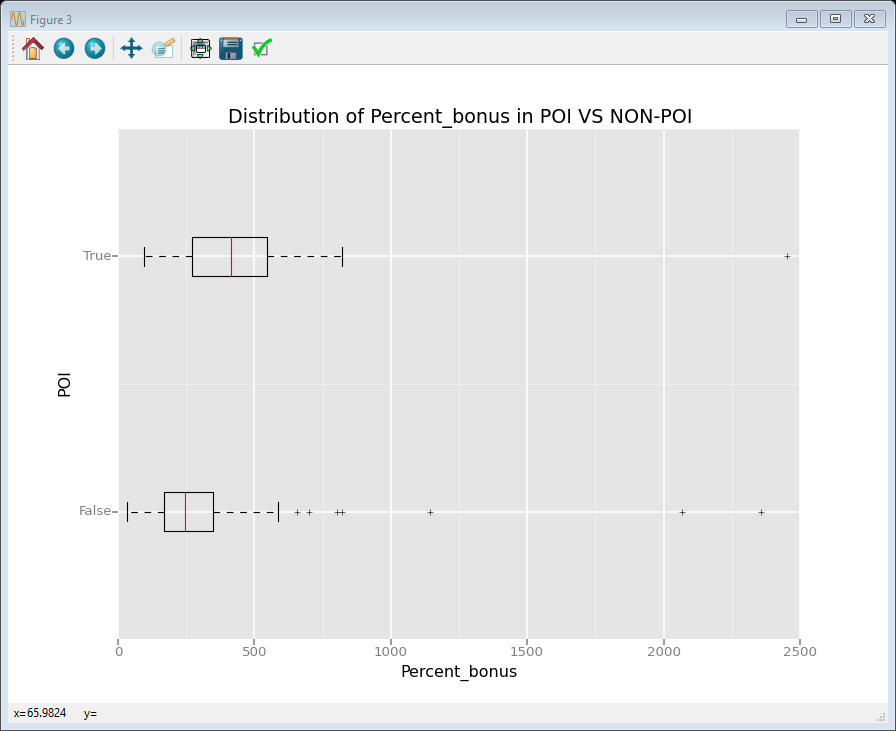
* **Percent\_bonus**=(**'bonus'**/**'salary'**)\*100
* **Ratio**\_**message** = **from\_this\_person\_to\_poi / from\_messages**
* **ratio\_to\_from\_messages= (from\_poi\_to\_this\_person + from\_this\_person\_to\_poi)/ to\_messages+from\_messages**

We will compare the impact of these new features in later sections.

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.

My initial selection of features was mostly based on the exploratory analysis that I have done in this section and my intuition. For example: From the box plots, I saw how salary, bonus, long term incentives etc are key characteristics to identify a poi.From the graphs, it was clear that these attribute play some important role and are significantly different for POIs than for NON POIs. Another key contribution came from the correlation matrix of the attributes and multivariate analysis, correlation matrix helped identifying the relationship among the attributes and I engineered new features for my model.

I also tried to use SelectKBest to select top best features for my model. I started with top 5 best features and went down up to 2 top best features for model. The comparison matrix for the selection is given in further section.

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”]

**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:

|  |  |
| --- | --- |
| **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\_from\_messages'** | Accuracy: 0.78075  Precision: 0.58554  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 |

**Impact of the features on the model:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Algorithm-GussianNB()** | | | |
| **1** | With new feature/s | [**'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 |
| Without new feature/s | **'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 |
| 2 | With new feature/s | **'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 |
| Without new feature/s | **'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 |
| 3 | With new feature/s | **'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 |
| Without new feature/s | **'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 |

|  |  |  |
| --- | --- | --- |
|  | **Algorithm-DecisionTreeClassifier** | |
| **Features** |  | **Result** |
| **'poi'**,**'salary'**,**'Percent\_bonus'**,**'long\_term\_incentive'**,**'exercised\_stock\_options'**,**'from\_poi\_to\_this\_person'** | **'salary': 0.42773752**  **'Percent\_bonus': 0.11574074**  **'long\_term\_incentive': 0.21875**  **'exercised\_stock\_options': 0.23777174**  **'from\_poi\_to\_this\_person':0.0** | 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'** | **'salary':** **0.31746032**  **'bonus':0.0**  **'long\_term\_incentive':0.0**  **'exercised\_stock\_options':0.0**  **'ratio\_messages': 0.68253968** | 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'** | **'salary':** **0.06728778**  **'bonus':** **0.21164021**  **'long\_term\_incentive':** **0.33449074**  **'exercised\_stock\_options':** **0.38658126**  **'from\_poi\_to\_this\_person':0.0** | 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'** | **'salary':** **0.16363636**  **'long\_term\_incentive':** **0.198**  **'exercised\_stock\_options':** **0.63836364**  **'ratio\_to\_from\_messages':** **0.0** | 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'** | **'salary':0**  **'long\_term\_incentive':** **0.198**  **'exercised\_stock\_options':** **0.802**  **'ratio\_to\_from\_messages':0** | 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']** | **'salary':** **0.33711934**  **'long\_term\_incentive':0.0**  **'total\_stock\_value':** **0.41636071'**  **ratio\_to\_from\_messages':** **0.2465199** | 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'** | **'Percent\_bonus':** **0.15873016 'long\_term\_incentive':0.0**  **'exercised\_stock\_options':** **0.14880952**  **'ratio\_messages': 0.69246032** | 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'** | **'Percent\_bonus':** **0.08928571**  **'long\_term\_incentive':** **0.54315476**  **'exercised\_stock\_options':** **0.36755952**  **'from\_poi\_to\_this\_person':0** | 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'** | **'bonus':** **0.14880952**  **'long\_term\_incentive':** **0.61341874**  **'exercised\_stock\_options':** **0.23777174**  **'from\_poi\_to\_this\_person':0** | 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 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Algorithm-Decision Tree classifier** | | | |
| **1** | With new feature/s | **'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 |
| Without new feature/s | **'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 |
| 2 | With new feature/s | **'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 |
| Without new feature/s | **'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 |
| Without new feature/s | **'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 |

**Select Kbest features :**

Below is the score for the key features:

* **'salary':** **5.46410678**
* **'bonus':** **3.89576806**
* **'to\_messages':** **0.16595733**
* **'total\_payments'**: 4.2923287
* **'exercised\_stock\_options':** **15.5445133**
* **'bonus':** **3.89576806**
* **'restricted\_stock':** **9.18669591**
* **'shared\_receipt\_with\_poi':** **0.07674135**
* **'total\_stock\_value':** **14.58481944**  
  **'from\_messages':** **0.49220496**
* **'from\_this\_person\_to\_poi':** **0.54997264  
  'long\_term\_incentive'**: 9.89786842
* **'from\_poi\_to\_this\_person'**: 0.30777798
* **'Percent\_bonus': 0.03766537**
* **'ratio\_messages':** **4.28545407**
* **'ratio\_to\_from\_messages'0.07329423**

|  |  |  |
| --- | --- | --- |
| **K** | **Features** | **Results** |
| 5 | **'salary'** ,**'exercised\_stock\_options'**, **'restricted\_stock'**, **'total\_stock\_value'**, **'long\_term\_incentive'** | Accuracy: 0.84300  Precision: 0.66117  Recall: 0.44100  F1: 0.52909 F2: 0.47247  Total predictions: 5000  True positives: 441  False positives: 226  False negatives: 559  True negatives: 3774 |
| 4 | ,**'exercised\_stock\_options'**, **'restricted\_stock'**, **'total\_stock\_value'**, **'long\_term\_incentive'** | Accuracy: 0.84100  Precision: 0.65140  Recall: 0.44100  F1: 0.52594 F2: 0.47146  Total predictions: 5000  True positives: 441  False positives: 236  False negatives: 559  True negatives: 3764 |
| 3 | **‘exercised\_stock\_options'**, **'total\_stock\_value'**, **'long\_term\_incentive'** | Accuracy: 0.83333  Precision: 0.50000  Recall: 0.42600  F1: 0.46004 F2: 0.43899  Total predictions: 6000  True positives: 426  False positives: 426  False negatives: 574  True negatives: 4574 |
| 2 | **‘exercised\_stock\_options'**, **'total\_stock\_value'**, | Accuracy: 0.89710  Precision: 0.48343  Recall: 0.42300  F1: 0.45120 F2: 0.43385  Total predictions: 10000  True positives: 423  False positives: 452  False negatives: 577  True negatives: 8548 |

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.

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? (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.

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 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. Inside tester.py , StratifiedShuffleSplit function is used, which is a combination of StratifiedKFold and ShuffleSplit. Stratified K-Folds provides train/test indices to split data in train test sets and shuffleSplit is permutation cross-validation iterator which yields indices to split data into training and test sets. StratifiedShuffleSplit is combination of these two functions. Also, in the tester.py Accuracy, Precision, Recall values are calculated by looping through the predicted values and true labels and counting the true positive, true negative, false positives and false negatives.Actual score is calculated based on the average of the score through each fold.

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”]

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 |