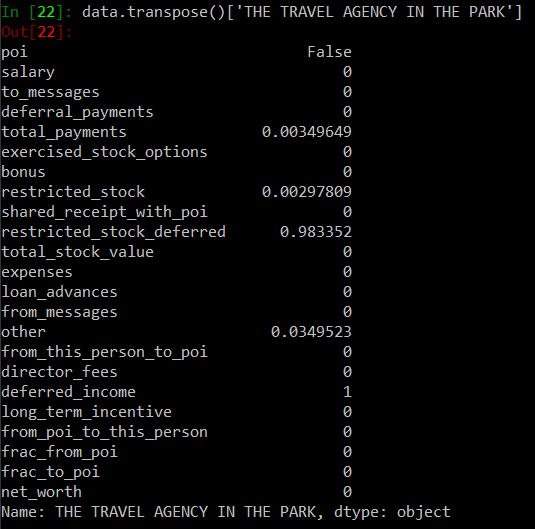
1. **Summarize for us the goal of this project and how machine learning is useful in trying to accomplish it. As part of your answer, give some background on the dataset and how it can be used to answer the project question. Were there any outliers in the data when you got it, and how did you handle those?  [relevant rubric items: “data exploration”, “outlier investigation”]**
   1. The goal of this project is to find a way to identify with high precision and recall the person-of-interest in the Enron dataset.
   2. Machine learning allows us to do this by using various metrics that “correlates” with the person-of-interest binary variable
   3. From the dataset, we can see that there are 146 observations
      1. Some important characteristics of the data
         1. There are a total of 146 data points
            1. Out of these 146, we identified “TOTAL” as an outlier as it is a summary statistics of the other 145 observations
            2. 'THE TRAVEL AGENCY IN THE PARK' does not seem to be a meaningful data point given that it has basically no data (as seen below). Thus it was removed as well to reduce noise in the dataset

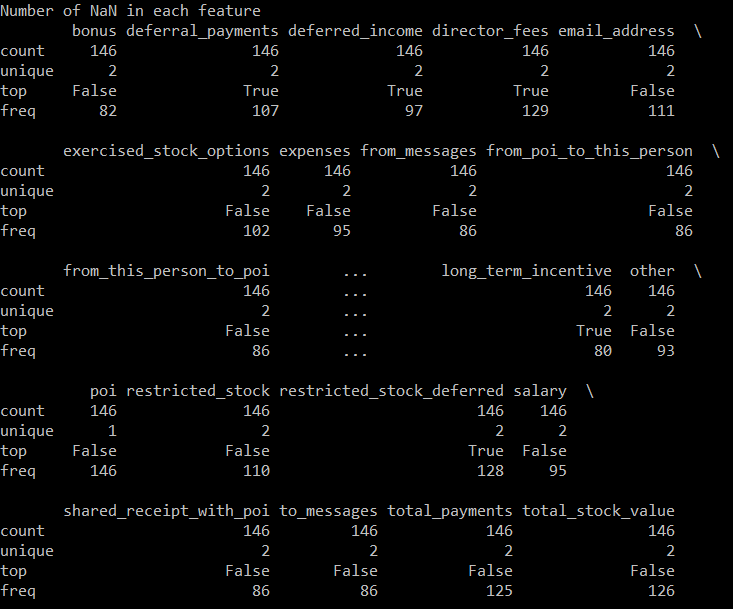


* + - 1. There are 18 POI out of the 145 observations
      2. A total of 21 features and 1 label is found in the data set
         1. This does not include any new variables created
         2. One of the feature is “email\_address” which is subsequently removed
         3. The features generally fall into 2 categories namely

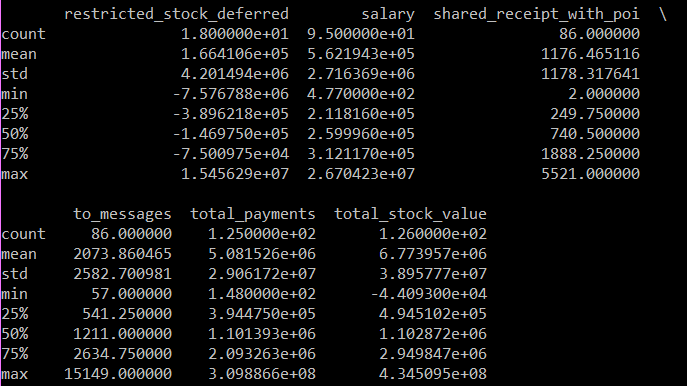
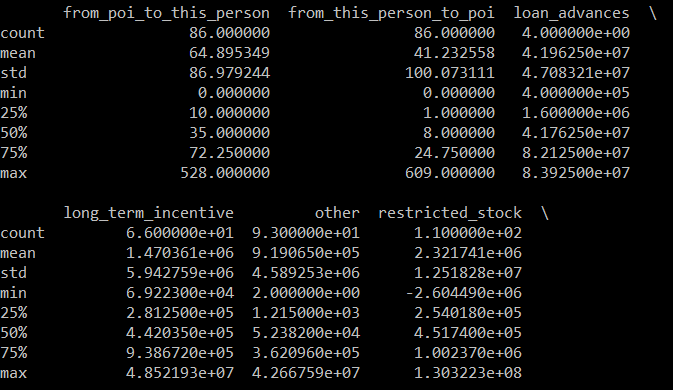
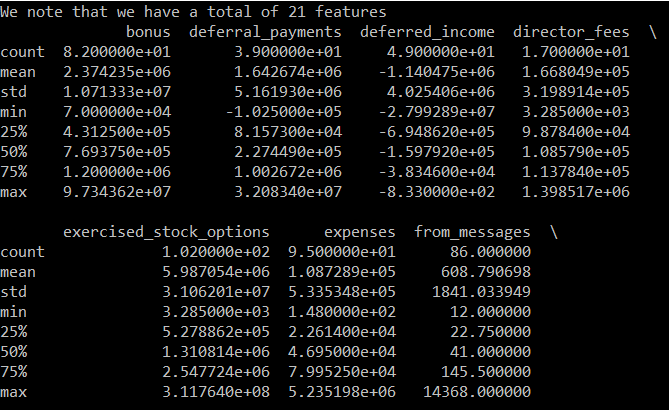
Economic incentive variables (e.g. “bonus”)

We note that most of the economic variables like director\_fees have a high amount of “NaN” in them

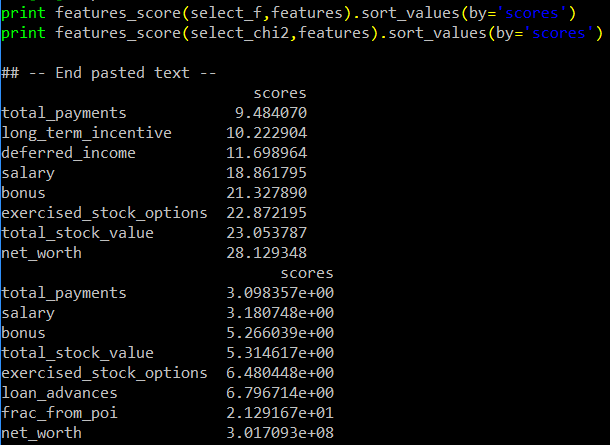
Frequency of contact variables (e.g. “from\_messages”) which tend to have less NaN.



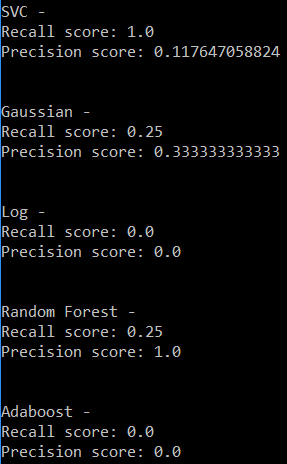
* + 1. Looking at the data manually, we note that 2 of the observations does not tie back out with the data on the PDF.
    2. Thus, I wrote a function “correct\_records” to correct the data for 'BELFER ROBERT' and 'BHATNAGAR SANJAY’
    3. Below we can see the summary statistics of the various measures in the dataset
       1. We note that the summary statistics are very varied in terms of scale.
       2. E.g. to\_messages are in thousands but total\_payments are in millions



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”]**
   1. I included all the various features in the initial analysis as I believe it will be presumptuous to start stripping out various measures
      1. However, email\_address as a feature was removed. It will not have any predictive ability.
   2. First I look to scale the measures using MinMaxScaler in sklearn.
      1. As noted in point 1, the measures are extremely varied where “total\_payments” is a factor of 1000 larger than “to\_messages”
      2. This makes various methods like gradient descent unstable
   3. Secondly, I look to include various measures that I believe are very useful
      1. For this purpose, I included 3 more features
         1. ‘frac\_from\_poi’ = 'from\_poi\_to\_this\_person' / 'from\_messages'
         2. 'frac\_to\_poi' = 'from\_this\_person\_to\_poi'/ 'to\_messages'
         3. 'net\_worth' = 'salary' + 'bonus' + 'total\_stock\_value
      2. ‘frac\_from\_poi’ and ‘frac\_to\_poi’ are deemed important because they show the extent of involvement this individual has with known person of interest.
      3. ‘net\_worth’, on the other hand, is to try to measure how well remunerated they are.
   4. After including these measures, I prepared a simple pipeline using LinearSVC and use it to decide the optimal number of features I should be choosing from these 23 features. The function tries out the performance given 8,12,16 and 20 features.
      1. Running the function “select\_features\_svc”, we note that the output suggested that 8 features is the optimal number of features.
   5. Lastly I used the SelectKBest function and set out with 2 selection criteria namely f class and chi-square so as to ensure that we get the best set of features:
      1. Below we notice that both tests selected a few common features that has significant predictive powers
         1. salary
         2. total\_payments
         3. exercised\_stock\_options
         4. bonus
         5. total\_stock\_value
         6. net\_worth
      2. However, the f test chose deferred\_income and long\_term\_incentive whereas chi2 chose loan\_advances and frac\_from\_poi.
         1. Looking at the score of long\_term\_incentive and deferred\_income, they seem rather low in the f test. Intuitively, I do feel that frac\_from\_poi and loan\_advance should feature more highly than both long\_term\_incentive and deferred\_income
         2. This is especially so when Enron is a classic case of fraud, deferred\_income and long\_term\_incentive should be the last thing on the mind of a POI who actually knows that.
      3. Thus, I decide to use the features selected by the chi-square test.



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”]**
   1. I used various metrics and algorithms to find the best estimator such that it has a good balance between accuracy and precision
      1. In the end, the two best estimators were RandomForestClassifier, which gave a recall score of 0.25 and a precision score of 1.0 and SVC which gave a recall score of 1.0 and precision score of 0.1176
   2. The various methods tested were
      1. Support Vector Machine
      2. Gaussian Classifier
      3. Logistic Regression Classifier
      4. RandomForestClassifier
      5. AdaBoostClassifier
   3. The results from the test were as follow



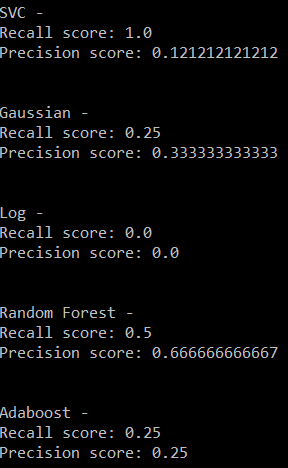
1. **What does it mean to tune the parameters of an algorithm, and what can happen if you don’t do this well?  How did you tune the parameters of your particular algorithm? (Some algorithms do not have parameters that you need to tune -- if this is the case for the one you picked, identify and briefly 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”]**
   1. Tuning the parameters of an algorithm is to change the control variables so as to allow it to perform better/ faster. If it is not done properly, we might end up with suboptimal results such as ending up at local minimal or having erratic results.
   2. I tuned my using the GridSearchCV function to see if there is such an increase in performance
      1. As seen in the screen shot below, we note that there is a slight improvement in terms of performance for the Linear SVC, Random Forest Classifier and the Adaboost.
         1. I tried tuning the LinearSVC by adjusting the penalty parameter C of the error term and also tried speeding it up by increasing the tolerance for stopping criteria. There was a marginal improvement in precision as compared to default settings and the estimator still manage to achieve perfect recall.
         2. Similarly for the Adaboost classifier, I attempted to tune it by adjusting the learning rate and in the end, it achieved a more respectable 0.25 for both recall and precision.
         3. The most significant improvement came from the Random Forest Classifier which has achieved a respectable 0.5 recall with a slight decrease in precision to 0.6667
            1. Specifically for the Random Forest Classifier, I attempted to tune the following 3 fields

n\_estimators using values, 3,5,10 or 15

max\_feature using “auto”, “sqrt”, “log2” or None

criterion using “gini” or “entropy”

* + - * 1. Given that Random Forest Classifier has the best f1 score (0.5\*0.66/(0.5+0.66)\*2) = 0.5689, I decide that Random Forest Classifier is the final chosen algorithm



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"]**
   1. We split the dataset we have into 2 separate sets namely a test sample set and a training sample set. Validation is the evaluation of the performance of the estimator, which has been tuned by the training sample set, against test sample set so we are able to know what is the prediction error associated with our estimator.
      1. A classic mistake that occurs is when we do cross validation is when the sample used for training and the sample used for validation are not randomly selected or have share the same samples.
         1. This occurs when the samples are sequential or due to coding error
         2. To avoid this issue, the best way is to use the GridSearchCV function which does cross validation within the sample in a random fashion and provide us with the average recall and precision of the samples
         3. If the test sample set is used for training the estimator, the estimator will tend to overfit and hence become unreliable in the real world application
      2. The test set is never used until the estimator is ready for testing.
         1. This separation of samples ensures that we do not overfit the data by having an independent testing sample set.
2. **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”]**
   1. The 2 evaluation metrics that was used for measuring the performance of the estimator is Recall and Precision
      1. Recall is also known as true positive rate
         1. It is the ratio of how many positives that are identified are true positives
         2. I.e. For our Random Forest Classifier, we identified 50% of all the True state correctly.
         3. For the case of our testing sample set, we correctly identified 2 of the 4 POIs out of the 44 samples
      2. Precision on the other hand is ratio of true positives over the sum of true positives and false positives.
         1. It shows the ratio of how accurate is the model when it predicts a positive.
         2. I.e. for our Random Forest Classifier, when it predicts a positive, it has a 66.67% probability of it being correct (True)
         3. For the case of our testing sample set, it means that every time the model states that an individual in the sample is a POI, there is 66.67% probability we are correct and that individual is indeed a POI.

References used:

<https://en.wikipedia.org/wiki/Precision_and_recall>

<http://scikit-learn.org/stable/documentation.html>