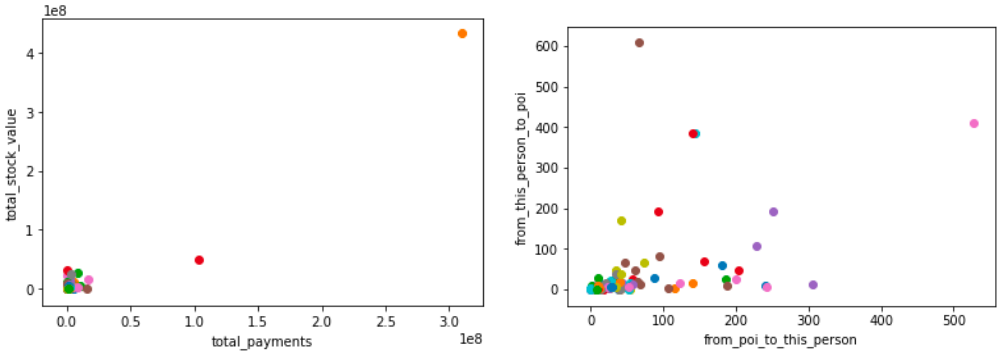
Enron Submission Free-Response Questions

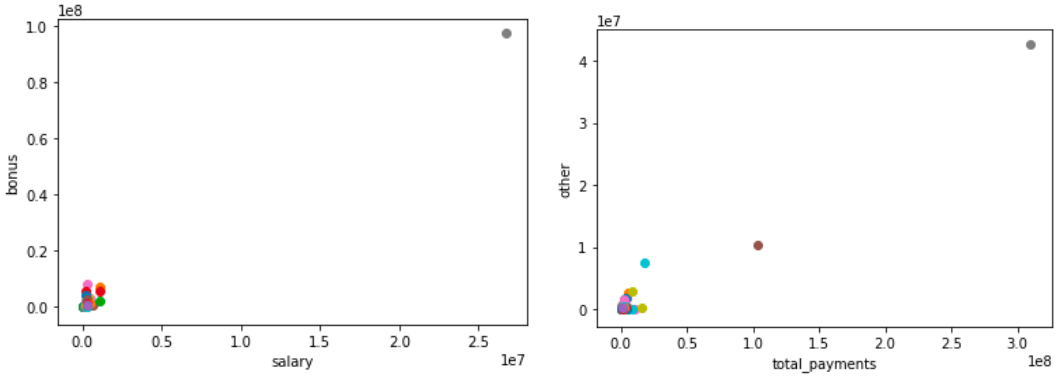
1. **Summarize for us the goal of this project and how machine learning is useful in trying to accomplish it. As part of your answer, give some background on the dataset and how it can be used to answer the project question. Were there any outliers in the data when you got it, and how did you handle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

   The goal of this project is to use the Enron dataset to build a prediction model to identifies person of interest. In the dataset we have 146 points, 18 of which are poi and 128 are non-poi.

   I noticed there are many 'NaN' entries in the dataset, so I investigate the number of missing values for each feature and here is the output:
   salary: 51
   to_messages: 60
   deferral_payments: 107
   total_payments: 21
   loan_advances: 142
   bonus: 64
   email_address: 35
   restricted_stock_deferred: 128
   total_stock_value: 20
   shared_receipt_with_poi: 60
   long_term_incentive: 80
   exercised_stock_options: 44
   from_messages: 60
   other: 53
   from_poi_to_this_person: 60
   from_this_person_to_poi: 60
   poi: 0
   deferred_income: 97
   expenses: 51
   restricted_stock: 36
   director_fees: 129

   For outliers, I created plots to visualize the data:





From the plots we can see there are some suspicious outliers, I tried to list out a few:
'TOTAL', 'LAY KENNETH L', 'FREVERT MARK A', 'BHATNAGAR SANJAY'

Here 'TOTAL' is apparently a mistaken entry in the dataset because it is a sum term, not a person. So I removed itfrom the dataset. On the other hand, the people weith outlier values may be a sign of poi so I don't want to remove those.

Also, after exploring variables that contains NaN features, I removed two more points in the dataset:
THE TRAVEL AGENCY IN THE PARK: This is not a person
LOCKHART EUGENE E: This record contains NaN only for the entire row

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", "intelligently select features", "properly scale features"]**

There are two new features I created named "fraction_from_poi" and "fraction_to_poi", which describe the ratio of the email send to/from a poi. I assume that this will improve the prediction because poi may have more frequent communication with one another.

For feature selection, I used SelectKBest to select the 10 features with highest scores. Here's the list of the score:

('exercised_stock_options', 24.815079733218194), ('total_stock_value', 24.182898678566879), ('bonus', 20.792252047181535), ('salary', 18.289684043404513), ('fraction_to_poi', 16.409712548035792), ('deferred_income', 11.458476579280369), ('long_term_incentive', 9.9221860131898225)

The final list of feature is: ['exercised_stock_options', 'total_stock_value', 'bonus', 'salary', 'fraction_to_poi', 'deferred_income', 'long_term_incentive']

I chose 7 as the parameter for SelectKBest after comparing the metrics of the test set below:

K=9
accuracy: 0.845348837209
precision: 0.372472222222
recall: 0.317576839827
K=8
precision: 0.408973665224
recall: 0.374167388167
K=7
precision: 0.431829365079
recall: 0.375834054834
K=6
precision: 0.449242063492
recall: 0.358794372294
K=5
precision: 0.450517316017
recall: 0.3185

At the same time, I realized that the numerical features have very wide range of values, while some of the classifier I tried to use, such as K-means, SVM and logistic regression, requires data to be scaled. So I min-max scaler to rescale each feature to a common range.

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"]**

I tried out four classifiers here: Naïve Bayes, K-Means, Logistic Regression and SVM. The result for each of them are showed in the following table:

Classifier	Accuracy	Precision	Recall	Best Parameter
Naïve Bayes	0.852619047619	0.431829365079	0.375834054834	-
K-Means	0.713571428571	0.467001625323	0.363350649351	N=2, tol=0.1
Logistic Regression	0.863333333333	0.507341269841	0.261941558442	tol=1, C=0.1
SVM	0.865714285714	0.154666666667	0.0600357142857	kernel = 'rbf', C = 0.1, gamma = 1

After considering all three metrics, I settled with Naïve Bayes model as it gives good result on each of them. I think K-Means also did a great job in identifying positive pois among all the algorithms.

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? 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"]**

Parameters will impact the performance of an algorithm. And the process of tuning parameters is to choose the optimal parameters that works properly with the data and thus gives the best performance the algorithm is capable of.

Without parameter tuning, the model may not perform at its optimal. But with too much parameter tuning, it may cause overfitting. So I want to be careful with it, and validation is also a good step to check it.

I used GridSearchCV to tune parameters and here is a list of best parameters I found for each algorithm:

Naïve Bayes: Tuning is not required

K-Means: N is set to be 2 because our target is binomial, tol=0.1

Logistic Regression: tol=1, C=0.1

SVM: kernel = 'rbf', C = 0.1, gamma = 1

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 items: "discuss validation", "validation strategy"]**

   Validation is a model validation technique that assesses how the results of a statistical analysis will generalize to an independent data set. A classic mistake is overfitting – in that case the model will fit the training set perfectly but cannot be generalize to other testing sets.

   Because the dataset is skewed, I applied the cross-validation technique by using StratifiedShuffleSplit to split the training and testing data 100 times and took the mean of each metric.

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"]**

   I used 3 evaluation metrics here:

   Accuracy: # of right predictions/ # of total data points. In this problem, this means the ratio of the people that have been correctly identified.

   Precision: # of true positives among predicted positives/ # of predicted positives. In this project, it is the ratio of real poi to those identified as a poi by our model.

   Recall: # of true positives predicted/ # of true positive. In this project, it gives the ratio of poi that have been successfully identified by our model.

   The performance of these metrics are shown in the table in Question 3.