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 purpose of this project is to apply classification machine learning algorithms to identify Person Of Interest (POI) using financial and email data from Enron.

   The data consist of the data consists of three types of features

   - Label:

   1- POI : flag weather the person is person of interest

   - Email features:

   2- email_address
   3- to_messages
   4- from_messages
   5- from_poi_to_this_person
   6- from_this_person_to_poi
   7- shared_receipt_with_poi

   - Finance:

   8- salary
   9- bonus
   10- long_term_incentive
   11- deferred_income
   12- deferral_payments
   13- loan_advances
   14- other
   15- expenses
   16- director_fees
   17- total_payments
   18- exercised_stock_options
   19- restricted_stock
   20- restricted_stock_deferred
   21- total_stock_value

   Original data rows (146) - Removed outlier (3) = 143

   Original features (21) – email address (1) + Engineered (2) = 22 (21 features+1 label)

   POIs =18
   Non-POIs = 125 (after removing outliers)

I identified 3 outliers for removal:

1- TOTAL: I have plot the data in scoter-plot, so it was easy to identify since it's extreme point. This seems a record form a spreadsheet that sum each financial data.
2- LOCKHART EUGENE E: this record has null values.
3- THE TRAVEL AGENCY IN THE PARK: not person

The reason why I removed the three outliers, because they are invalid data points and I don't want it to affect the result of the classification. I have found some outlier but they were true data point so I decided to keep them since they are a valid data points. Such as (Kenneth lay, Jeffrey Skilling).

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"]

Feature selection:

In the feature selection processes, I have tried both all features with the new two features and I also used SelectKBest(k=11) I choose the k=11 since I have 22 features, which is see it's best to divide the features. In addition, I approached a way to settle on the selection of k=11 is by test the evaluation of SVM for the possible K values and high scores features listed below.

Rank	**feature**	**Score**
1	'exercised_stock_options'	24.81507973
2	'total_stock_value'	24.18289868
3	'bonus'	20.79225205
4	'salary'	18.28968404
5	'fraction_to_poi'	16.40971255
6	'deferred_income'	11.45847658
7	'long_term_incentive'	9.92218601

8	'restricted_stock'	9.21281062
9	'total_payments'	8.77277773
10	‘shared_receipt_with_poi’	8.58942073
11	‘loan_advances’	7.18405566

Keeping the POI lable in all the list

K=k	**Evaluation of SVM algorithm**
K=2 ('salary', 'total_payments')	validation#2 precision is 0.666666666667 validation#2 recalll is 0.3 validation#2 Accuracy is 0.869230769231
K=3 ('salary', 'total_payments','loan_advances')	validation#2 precision is 0.666666666667 validation#2 recalll is 0.3 validation#2 Accuracy is 0.869230769231
K=4 'salary', 'total_payments', 'loan_advances','bonus',	validation#2 precision is 0.44 validation#2 recalll is 0.55 validation#2 Accuracy is 0.823076923077
K=5 ('salary', 'total_payments','loan_advances','bonus', 'total_stock_value',)	validation#2 precision is 0.833333333333 validation#2 recalll is 0.25 validation#2 Accuracy is 0.893333333333
K=6 'salary', 'total_payments', 'loan_advances' 'bonus', 'total_stock_value', 'shared_receipt_with_poi',	validation#2 precision is 0.333333333333 validation#2 recalll is 0.85 validation#2 Accuracy is 0.753333333333

K=7 'salary', 'total_payments', 'loan_advances' 'bonus', 'total_stock_value', 'shared_receipt_with_poi', 'exercised_stock_options',	validation#2 precision is 0.34 validation#2 recalll is 0.85 validation#2 Accuracy is 0.76
K=8 'salary', 'total_payments', 'loan_advances' 'bonus', 'total_stock_value', 'shared_receipt_with_poi', 'exercised_stock_options', 'deferred_income',	validation#2 precision is 0.309090909091 validation#2 recalll is 0.85 validation#2 Accuracy is 0.726666666667
K=9 'salary', 'total_payments', 'loan_advances' 'bonus', 'total_stock_value', 'shared_receipt_with_poi', 'exercised_stock_options', 'deferred_income', 'restricted_stock',	validation#2 precision is 0.309090909091 validation#2 recalll is 0.85 validation#2 Accuracy is 0.726666666667
K=10 'salary', 'total_payments', 'loan_advances' 'bonus', 'total_stock_value', 'shared_receipt_with_poi', 'exercised_stock_options', 'deferred_income', 'restricted_stock', 'long_term_incentive',	validation#2 precision is 0.269230769231 validation#2 recalll is 0.7 validation#2 Accuracy is 0.706666666667
K=11 'salary', 'total_payments', 'loan_advances' 'bonus', 'total_stock_value', 'shared_receipt_with_poi', 'exercised_stock_options', 'deferred_income', 'restricted_stock', 'long_term_incentive', 'fraction_to_poi'	validation#2 precision is 0.38 validation#2 recalll is 0.95 validation#2 Accuracy is 0.786666666667

Listing the high scored features and testing the evaluation of the algorithm using different k values. I concluded that K=11 gives high evaluation value for both Precision&Recall metrics.

Scaling:

I have scaled the features using **(MinMaxScaler)** that scale all the features to be between 0 and 1 due to numeric values in both the finance data and email data in different scales. Specially finance data have large ranges. Also I used this scale because it has a an impact for some algorithms like SVM which I used in identifying the POI.In General, features scaling that before using the classifier all the features would be weighted evenly.

Feature engineering

I was looking to engineer a feature that related to POI. Emails communications is a key for identifying poi. So I decided to see the ratio of from/to poi to the total of to/from messages. This will be helpful to see the person send or receive email to/from poi by how much considering the totals messages.

```
fraction_to_poi = data_point["from_this_person_to_poi"] / data_point["from_messages"]
fraction_from_poi = data_point["from_poi_to_this_person"]/ data_point["to_messages"]
```

here I will justify the improvement of the evaluation (Recall and Precision) when adding the engineered feature:

```
[ 18.28968404  20.79225205   0.22461127  11.45847658   2.1263278
  24.81507973   6.09417331   7.18405566   9.92218601   4.18747751
   9.21281062   0.06549965   8.77277773  24.18289868   0.16970095
   1.64634113   5.24344971   2.38261211   8.58942073   3.12809175
  16.40971255]
['poi', 'salary', 'bonus', 'deferral_payments', 'deferred_income', 'director_fees', '
exercised_stock_options', 'expenses', 'loan_advances', 'long_term_incentive', 'other'
, 'restricted_stock', 'restricted_stock_deferred', 'total_payments', 'total_stock_val
ue', 'from_messages', 'to_messages', 'from_poi_to_this_person', 'from_this_person_to_
poi', 'shared_receipt_with_poi', 'fraction_from_poi', 'fraction_to_poi']
11 best features: ['salary', 'total_payments', 'loan_advances', 'bonus', 'total_stock
_value', 'shared_receipt_with_poi', 'fraction_to_poi', 'exercised_stock_options', 'de
ferred_income', 'restricted_stock', 'long_term_incentive']
```

from the above screenshot, we can see the score of the added features:

Feature	Score
'fraction_to_poi'	16.40971255
'fraction_from_poi'	3.12809175

So, I decided to add ‘fraction_to_poi’ to features list and Ignore ‘fraction_from_poi’ for now

Evaluation before adding ‘fraction_to_poi’	Evaluation after adding ‘fraction_to_poi’’
validation#2 precision is 0.269230769231 validation#2 recalll is 0.7 validation#2 Accuracy is 0.706666666667	validation#2 precision is 0.38 validation#2 recalll is 0.95 validation#2 Accuracy is 0.786666666667

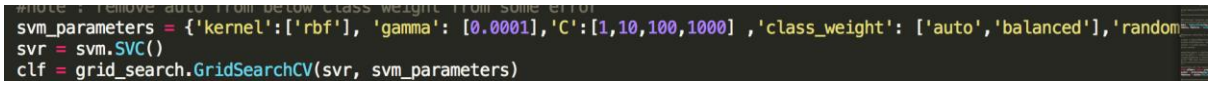
It’s obvious that adding ‘fraction_to_poi’ add up to the evaluation, both recall and precision improved.

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”]

Used algorithms:

1- Support vector machine
2- RandomForest
3- DecisionTree

I ended up using the svm classifier since in the evaluation it gives better performance. Initially, I tried all the features before updating the feature list that based on the SelectKBest and after updated the features based on it. The result For SVM as below




```
best paramters
{'kernel': 'rbf', 'C': 1000, 'random_state': 42, 'gamma': 0.0001, 'class_wei
ght': 'auto'}
```

```
best paramters
{'kernel': 'rbf', 'C': 1000, 'random_state': 42, 'gamma': 0.0001, 'class_weight': 'au
to'}
validation#1 prediction is
[ 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  1.  0.  0.  0.  0.  0.  0.
  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
  1.  1.  0.  0.  0.  0.  1.]
validation#1 precision is
0.25
validation#1 recalll is
0.2
validation#1 Accuracy is
0.837209302326
/Users/hana/anaconda2/lib/python2.7/site-packages/sklearn/utils/class_weight.py:65: D
eprecationWarning: The class_weight='auto' heuristic is deprecated in 0.17 in favor o
f a new heuristic class_weight='balanced'. 'auto' will be removed in 0.19
  " 0.19", DeprecationWarning)
validation#2 prediction is
[ 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  1.  1.  0.  0.  1.  1.  1.  0.
  1.  1.  1.  0.  0.  0.  0.  0.  0.  0.  1.  0.  0.  0.  0.  0.  0.  1.
  1.  0.  0.  0.  1.  1.  0.  0.  0.  0.  0.  0.  0.  0.  1.  0.  0.  0.
  0.  0.  1.  1.  0.  0.  0.  1.  1.  0.  1.  1.  1.  1.  0.  0.  0.  0.
  0.  1.  1.  0.  0.  0.  0.  0.  0.  1.  0.  1.  1.  0.  0.  1.  0.  0.
  1.  1.  0.  0.  1.  0.  1.  1.  1.  0.  1.  0.  0.  1.  0.  0.  0.  0.
  0.  0.  1.  0.  1.  0.  1.  0.  0.  0.  0.  1.  1.  0.  0.  1.  0.  0.
  0.  0.  1.  0.  0.  0.  1.  0.  1.  0.  1.  1.  0.  0.  1.  0.  0.  0.
  0.  0.  1.  0.  0.  1.]
validation#2 precision is
0.38
validation#2 recalll is
0.95
validation#2 Accuracy is
0.786666666667
```

I can see the different in evaluation value for the StratifiedShuffleSplit validation the precision&recall is better values. So I decided to complete the other algorithms with high scores.

The other two algorithms: RandomForest

```
best paramters
{'random_state': 42, 'criterion': 'entropy', 'class_weight': 'balanced'}
validation#1 prediction is
[ 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
  0.  0.  0.  0.  0.  0.  0.]
validation#1 precision is
/Users/hana/anaconda2/lib/python2.7/site-packages/sklearn/metrics/classifica
tion.py:1113: UndefinedMetricWarning: Precision is ill-defined and being set
 to 0.0 due to no predicted samples.
  'precision', 'predicted', average, warn_for)
0.0
validation#1 recalll is
0.0
validation#1 Accuracy is
0.883720930233
validation#2 prediction is
[ 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  1.  0.  0.  0.  1.  0.  0.  0.
  0.  1.  0.  0.  0.  0.  0.  0.  0.  0.  1.  0.  0.  0.  0.  0.  0.  0.
  0.  0.  0.  0.  1.  1.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
  0.  0.  1.  1.  0.  0.  0.  0.  1.  0.  1.  0.  0.  0.  0.  0.  0.  0.
  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  1.  0.  0.  0.  1.  0.  0.
  0.  0.  0.  0.  0.  0.  1.  0.  1.  0.  0.  0.  0.  0.  0.  0.  0.  0.
  0.  0.  0.  0.  0.  0.  0.  1.  0.  0.  0.  1.  1.  0.  0.  0.  0.  0.
  0.  0.  1.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
  0.  0.  1.  0.  0.  1.]
validation#2 precision is
1.0
validation#2 recalll is
1.0
validation#2 Accuracy is
1.0
```

DecisionTree:

```
best paramters
{'min_samples_split': 20, 'splitter': 'random', 'random_state': 13, 'criteri
on': 'gini', 'class_weight': 'balanced'}
validation#1 prediction is
[ 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  1.  0.  0.  1.  1.  0.  1.
  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  1.  1.  1.  0.  0.  0.  0.  0.
  1.  1.  0.  1.  0.  0.  1.]
validation#1 precision is
0.272727272727
validation#1 recalll is
0.6
validation#1 Accuracy is
0.767441860465
validation#2 prediction is
[ 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  1.  1.  1.  0.
  1.  1.  1.  0.  0.  0.  0.  0.  0.  0.  0.  1.  0.  0.  0.  0.  0.  0.
  0.  1.  0.  0.  1.  1.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  1.  0.
  0.  0.  1.  1.  0.  0.  0.  0.  1.  0.  1.  1.  1.  0.  1.  0.  0.  0.
  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  1.  0.  0.  0.  1.  0.  0.
  1.  0.  0.  0.  1.  0.  1.  1.  1.  0.  0.  0.  0.  0.  0.  0.  0.  0.
  0.  0.  0.  0.  0.  0.  0.  1.  0.  0.  0.  1.  1.  0.  0.  1.  0.  0.
  0.  0.  1.  0.  0.  0.  0.  0.  0.  0.  0.  1.  1.  0.  1.  0.  0.  0.
  0.  0.  1.  0.  0.  0.]
validation#2 precision is
0.529411764706
validation#2 recalll is
0.9
validation#2 Accuracy is
0.88
```

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 item: "tune the algorithm"]

   Tuning the parameters is the process of changing the parameters of the algorithm to find the best result. And by the best result I mean gives the better performance. There are two type of tuning the parameters one is manually changing it or using GridSearchCV to automate the process.

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"]

   Validation in Machine learning is done to see if the algorithm generalizes well. Since the known mistakes in validating the classification is over-fitting which means the model performed well in the training dataset, but not on the testing dataset.

   I tried two validation methods for validating the model as below:

   1- Split the data into train/test dataset
   2- KFlod/ShuffleSplit: This method has advantage above the first one because it allows use of all the data to train and test the model. Technically I used StratifiedShuffleSplit which returns stratified randomized folds. It'd a merge of StratifiedKFold and ShuffleSplit.it's split the data into 10 in my case and reuns the POI Identifier on each of these sets.

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 have utilized two evaluation method/metric Precision&Recall for testing the classifier. Although I have print the accuracy but I did not relay on it in choosing the algorithms and best parameters. Since the non-POIs is more than the POIs and if 'non-POI' had been predicted for all the records the accuracy would be high.

   Precision is the ratio of how often your model is correct in identifying a positive label (POI=1/true) to the total times it guesses a positive label. When we get high precision score this means less false positives.
   Recall is the ratio of how often your model correctly identifies a label as positive (POI =1/true) to how many positive labels there. When we get high recall, score would mean less false negatives.

   In my chosen algorithms (SVM) I got 0.35 Precision and 0.95 recall, I can say that the model predicted 35% of employee are POI correctly. Also 95% of recall means the model predicted this percentage of POI persons in the entire dataset

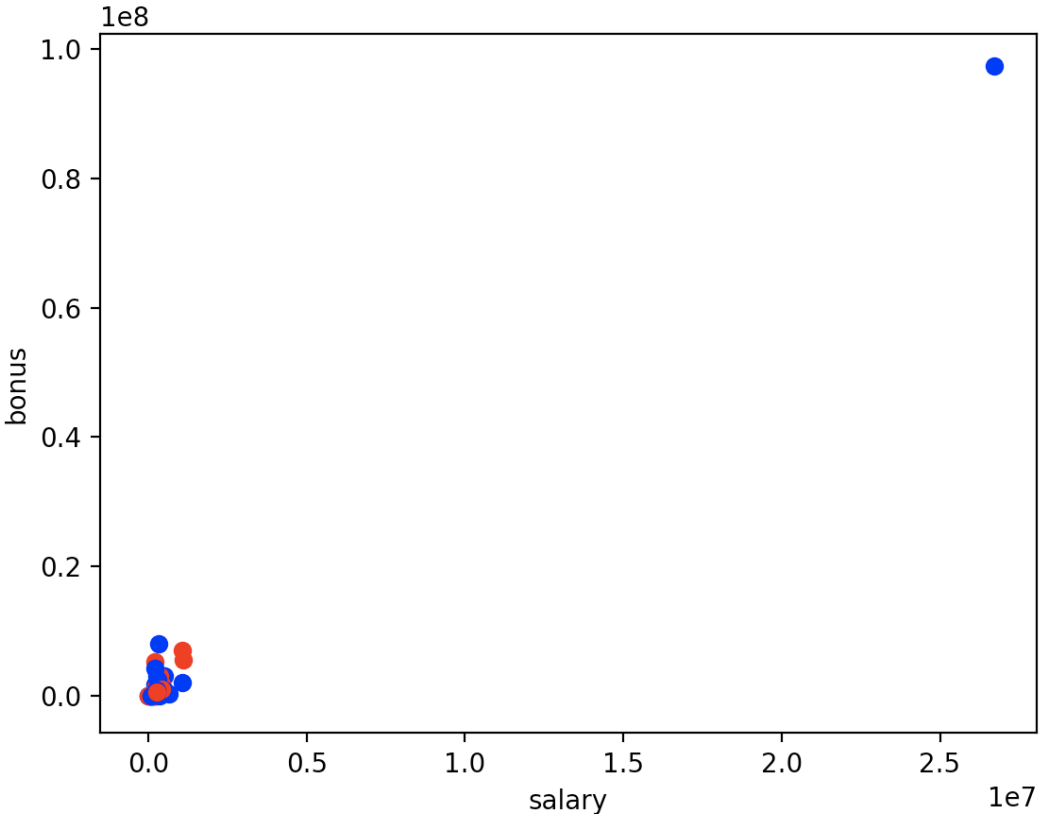


Figure 1 total outlier

Reference:

http://scikit-learn.org/stable/modules/feature_selection.html
https://www.quora.com/How-do-I-properly-use-SelectKBest-GridSearchCV-and-cross-validation-in-the-sklearn-package-together
https://www.youtube.com/watch?v=Gol_qOgRqfA&t=22s
https://www.civisanalytics.com/blog/workflows-in-python-using-pipeline-and-gridsearchcv-for-more-compact-and-comprehensive-code/
http://scikit-learn.org/stable/modules/generated/sklearn.cross_validation.StratifiedShuffleSplit.html