Identity Fraud From Enron Email

by Esther Xu
September 2017

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”]

1.1 Background

The Enron story is the largest case of corporate fraud in American history.Enron was an energy trading company.In 2000, Enron was one of the largest companies in the United States. By 2002, it had collapsed into bankruptcy due to widespread corporate fraud.

1.2 Objective

This project is to use machine learning skills to identify Enron Employees who may have committed fraud based on the public Enron financial and email dataset.
In other word,this project is to come up with a predictive patterns in the emails of persons of interest(POIs) who was charged with a crime,settled without admitting guilt,or testified for the government in exchange for immunity from prosecution.

1.3 Data Understanding

- **How many data do we have? How many features in this dataset?**
  We are working with 146 observations (persons) of 21 features, And 18 POIs are in the dataset.

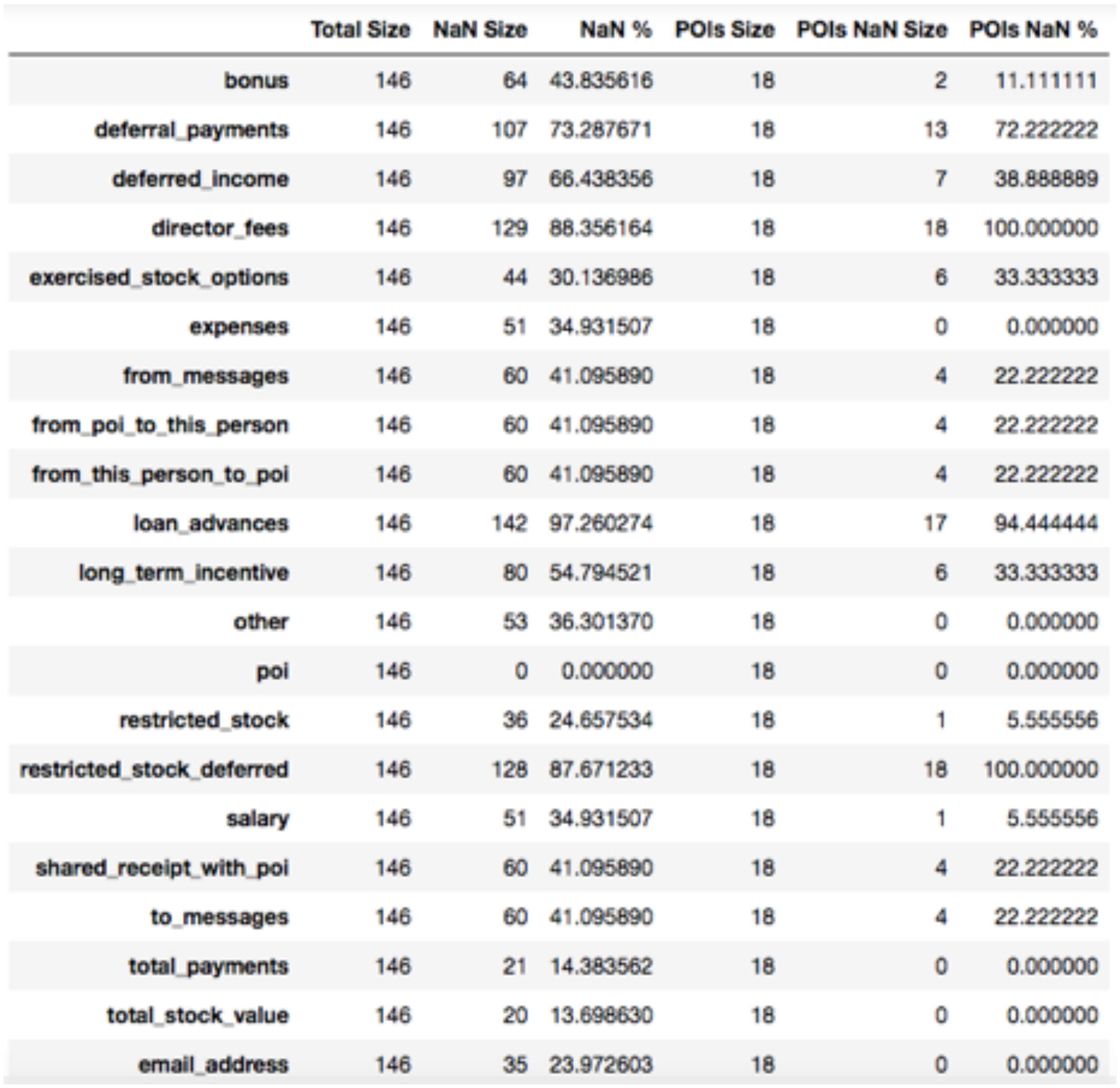
- **Which features are available in the dataset?**
  The following are three major types:
  financial features
  ['salary', 'deferral_payments', 'total_payments', 'loan_advances', 'bonus', 'deferred_income', 'total_stock_value', 'expenses', 'exercised_stock_options', 'other', 'long_term_incentive', 'restricted_stock_deferred','restricted_stock', 'director_fees'] (all units are in US dollars)
  email features
  ['to_messages', 'email_address', 'from_poi_to_this_person', 'from_messages', 'from_this_person_to_poi', 'shared_receipt_with_poi'] (units are generally number of emails messages; notable exception is ‘email_address’, which is a text string)
  person of interest label(poi)
  [‘poi’] (boolean, represented as integer)

- **Which features are categorical?**
  These values classify the samples into sets of similar samples. Within categorical features are the values nominal, ordinal, ratio, or interval based?
  Categorical: ['poi']

- **Which features are numerical?**
  These values change from sample to sample. Within numerical features are the values discrete,continuous, or timeseries based?
  Discrete: ['salary', 'deferral_payments', 'total_payments', 'loan_advances', 'bonus', 'restricted_stock_deferred', 'deferred_income', 'total_stock_value', 'expenses', 'exercised_stock_options', 'other', 'long_term_incentive', 'restricted_stock', 'director_fees','to_messages', 'from_poi_to_this_person', 'from_messages', 'from_this_person_to_poi', 'shared_receipt_with_poi']

- **Which features are mixed data types?**
  Numerical, alphanumeric data within same feature. These are candidates for correcting goal.
  email_address is alphanumeric.

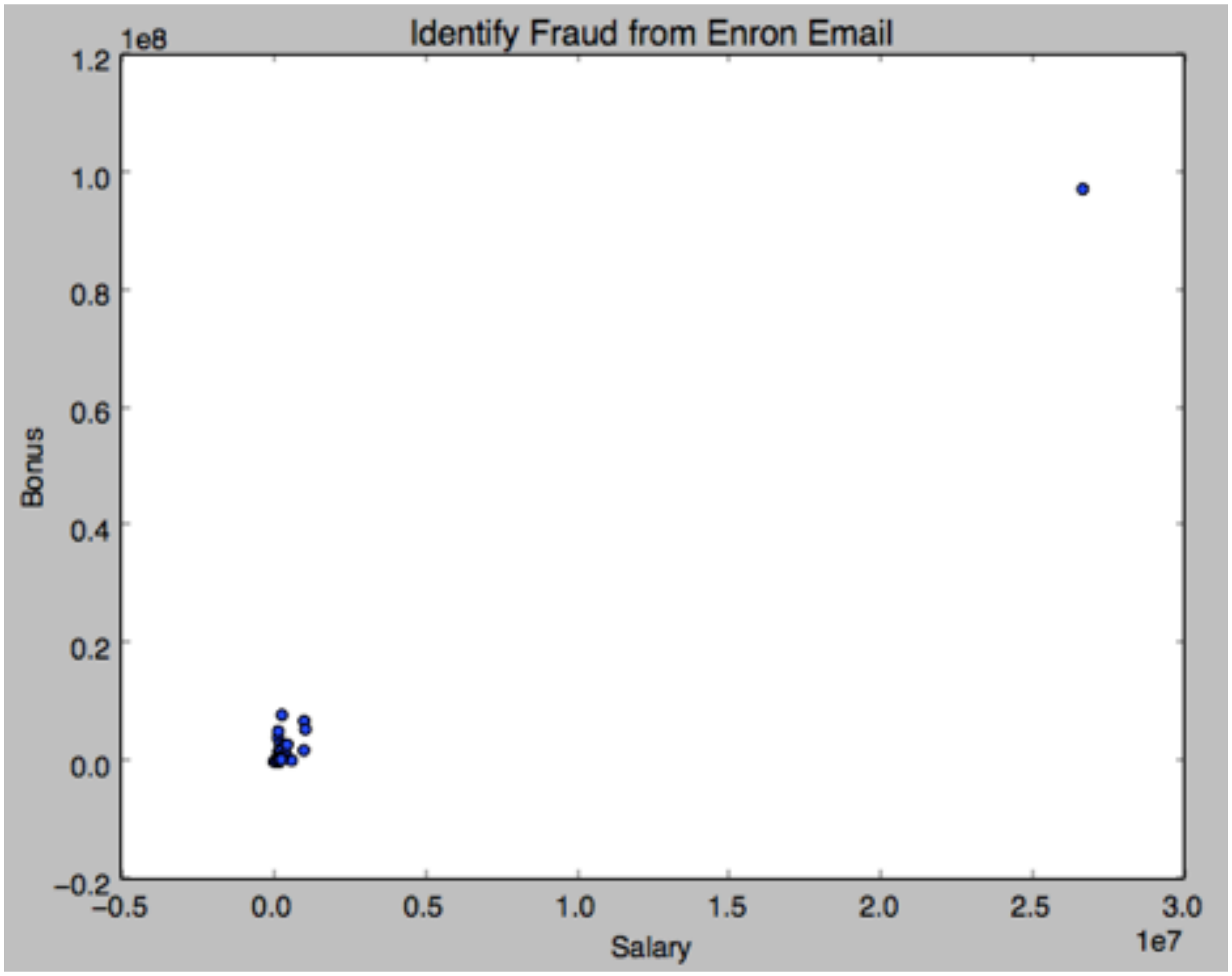
- **Which features contain blank, null or empty values? and Which percentage of POIs in the dataset have 'NaN' for each features?**
  I use util.get_illustration_4_NaN(data_df) function which I create for checking the missing value for each feature. You can we



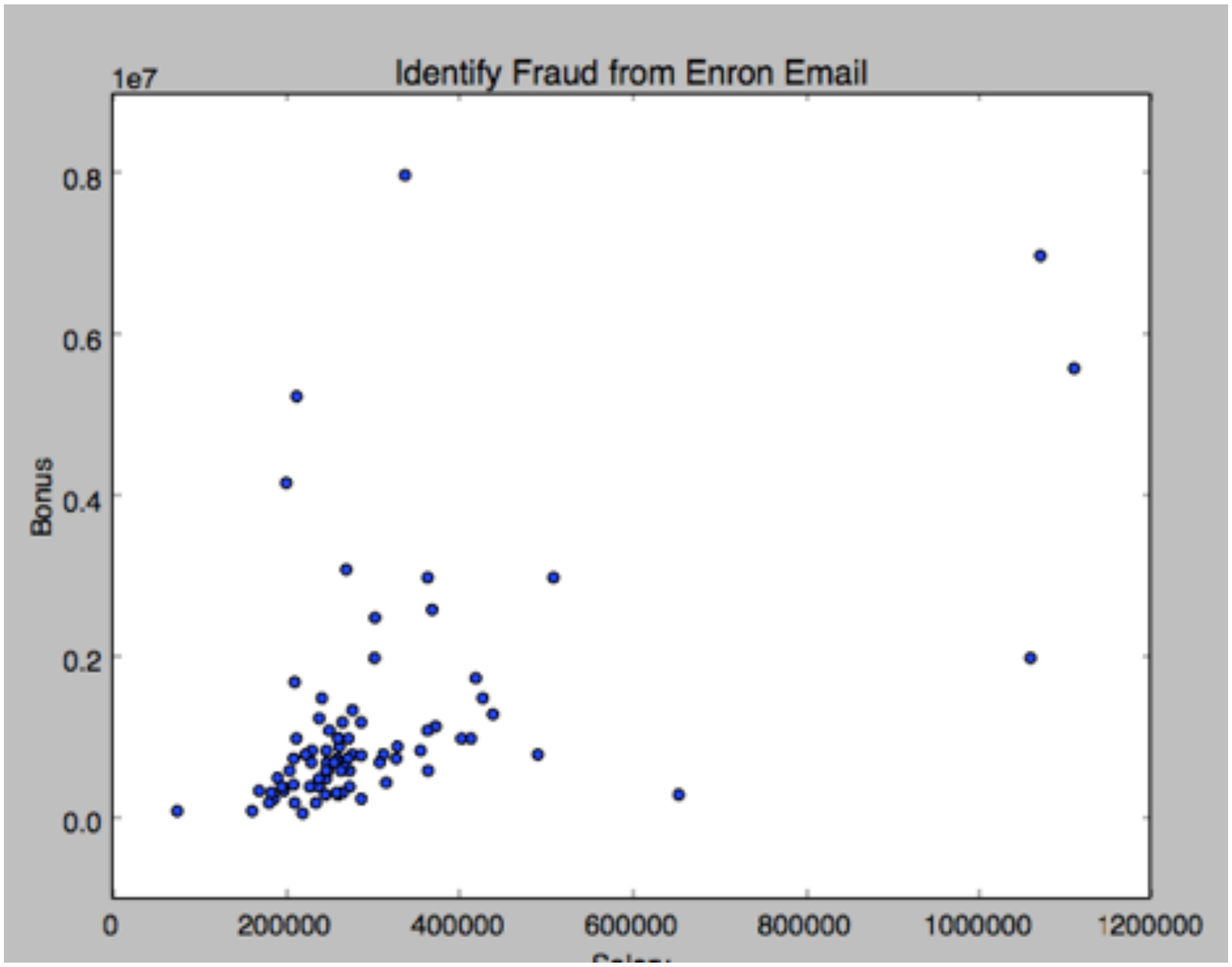
	Total Size	NaN Size	NaN %	POIs Size	POIs NaN Size	POIs NaN %
bonus	146	64	43.835616	18	2	11.111111
deferral_payments	146	107	73.287671	18	13	72.222222
deferred_income	146	97	66.438356	18	7	38.888889
director_fees	146	129	88.356164	18	18	100.000000
exercised_stock_options	146	44	30.136986	18	6	33.333333
expenses	146	51	34.931507	18	0	0.000000
from_messages	146	60	41.095890	18	4	22.222222
from_poi_to_this_person	146	60	41.095890	18	4	22.222222
from_this_person_to_poi	146	60	41.095890	18	4	22.222222
loan_advances	146	142	97.260274	18	17	94.444444
long_term_incentive	146	80	54.794521	18	6	33.333333
other	146	53	36.301370	18	0	0.000000
poi	146	0	0.000000	18	0	0.000000
restricted_stock	146	36	24.657534	18	1	5.555556
restricted_stock_deferred	146	128	87.671233	18	18	100.000000
salary	146	51	34.931507	18	1	5.555556
shared_receipt_with_poi	146	60	41.095890	18	4	22.222222
to_messages	146	60	41.095890	18	4	22.222222
total_payments	146	21	14.383562	18	0	0.000000
total_stock_value	146	20	13.698630	18	0	0.000000
email_address	146	35	23.972603	18	0	0.000000

1.4 Data Preparation

- **Outlier**



Through visualizing using box-plot,there's an extreme outlier called 'TOTAL' which should be total of numerical features that every person in dataset,but counted as a person. We should remove it because it's not a data that we have attention.



There are still have 4 outlier that identified as POI after removing.Since this is the data that we have to pay attention, we can't remove the outlier.

Since having 'TOTAL' data in the dataset. Do we have the other data entry errors for person's name? Person name is formated like last name + space + first name + space + middle name(if exists),In other word,it is less than three and greater than one space. Apparently,'THE TRAVEL AGENCY IN THE PARK' is not a real name.We should remove it.

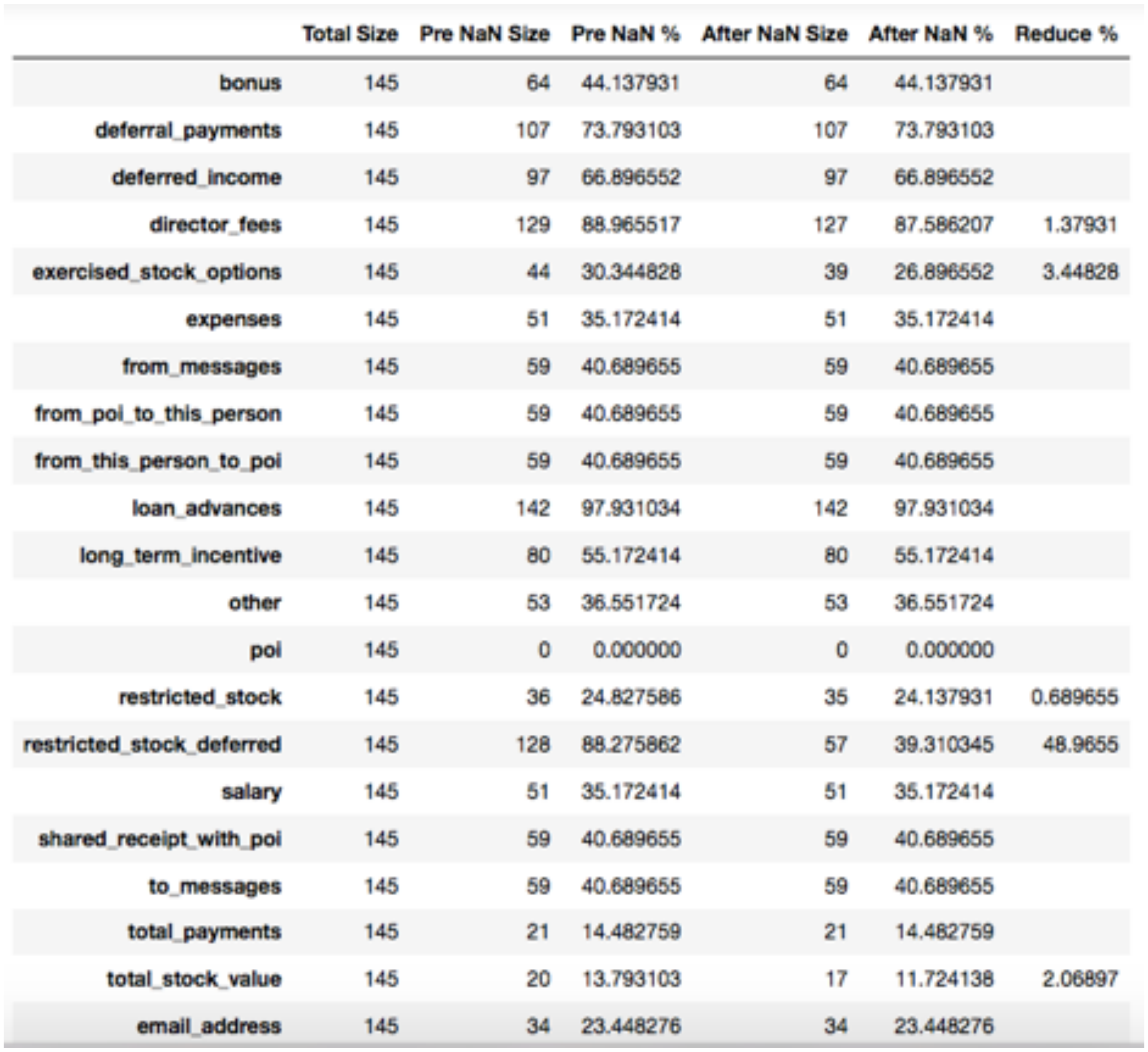
According to null illustration, all features have null values except 'poi' feature.We also have a person named ‘LOCKHART EUGENE E’ who has null values of all features in section of numerical features and is not POI.
This record is not help us to analyze,it should be removed.

- **Categorical Variables Transformed**

The poi is treated as categorical variables.Some of our model algorithms can only handle numeric values and so we should create a new variable (dummy variables) or update for every unique value of the categorical variables.

- **Missing values imputation**

I use two ways to impute which is formula and RandomForest algorithm.After missing values process,illustrating there are missing values of some feature are improved.We may validate these assumptions further before taking appropriate actions.



	Total Size	Pre NaN Size	Pre NaN %	After NaN Size	After NaN %	Reduce %
bonus	145	64	44.137931	64	44.137931	
deferral_payments	145	107	73.793103	107	73.793103	
deferred_income	145	97	66.896552	97	66.896552	
director_fees	145	129	88.965517	127	87.586207	1.37931
exercised_stock_options	145	44	30.344828	39	26.896552	3.44828
expenses	145	51	35.172414	51	35.172414	
from_messages	145	59	40.689655	59	40.689655	
from_poi_to_this_person	145	59	40.689655	59	40.689655	
from_this_person_to_poi	145	59	40.689655	59	40.689655	
loan_advances	145	142	97.931034	142	97.931034	
long_term_incentive	145	80	55.172414	80	55.172414	
other	145	53	36.551724	53	36.551724	
poi	145	0	0.000000	0	0.000000	
restricted_stock	145	36	24.827586	35	24.137931	0.689655
restricted_stock_deferred	145	128	88.275862	57	39.310345	48.9655
salary	145	51	35.172414	51	35.172414	
shared_receipt_with_poi	145	59	40.689655	59	40.689655	
to_messages	145	59	40.689655	59	40.689655	
total_payments	145	21	14.482759	21	14.482759	
total_stock_value	145	20	13.793103	17	11.724138	2.06897
email_address	145	34	23.448276	34	23.448276	

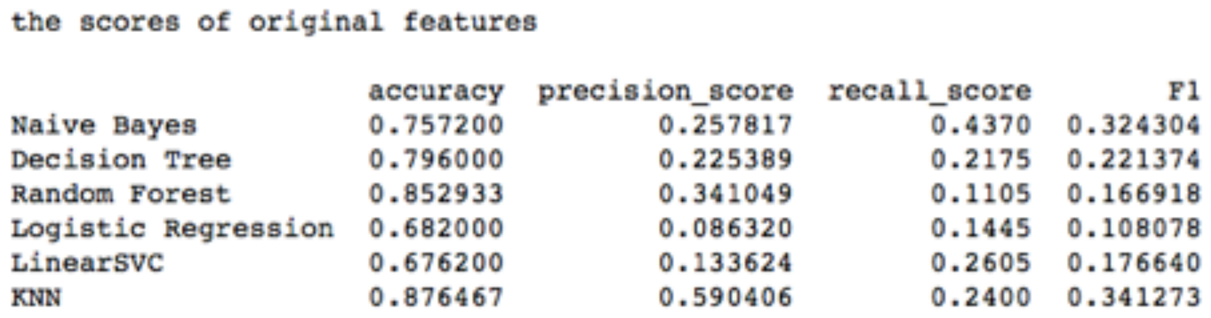
- 'deferral_payments','director_fees','loan_advances' features may be dropped from our analysis as there are highly contains many null values in dataset. If we add these features which have highly missing value,they will be noise.
- The rest of missing values may be zero in reality.

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"]

2.1 Create new feature combining existing features

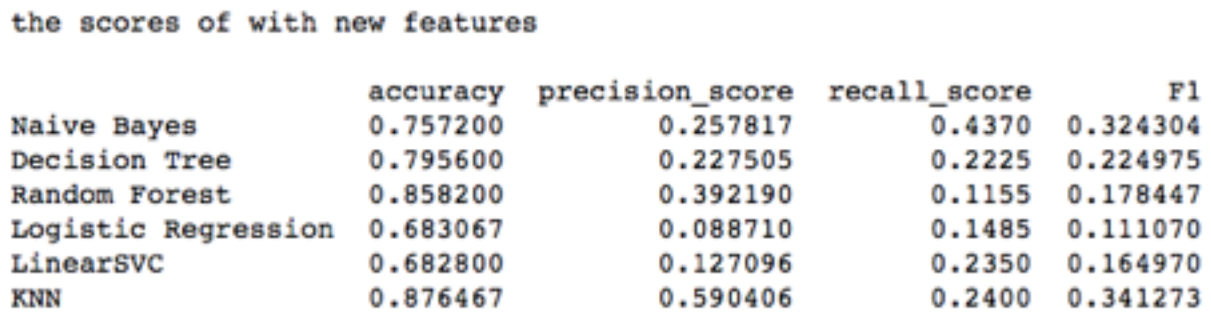
If this person have higher frequencies of sending and receiving email with POIs, this person could end up being POI himself. So I create the fraction of new features in which this person sent email to POIs and received from POIs.

Compared the performance of each algorithm with original best features(excluding new features) to with adding the new features(including new features), to observe the effect of adding my new engineered features.



```
the scores of original features

                     accuracy  precision_score  recall_score        F1
Naive Bayes          0.757200         0.257817        0.4370  0.324304
Decision Tree        0.796000         0.225389        0.2175  0.221374
Random Forest        0.852933         0.341049        0.1105  0.166918
Logistic Regression  0.682000         0.086320        0.1445  0.108078
LinearSVC            0.676200         0.133624        0.2605  0.176640
KNN                  0.876467         0.590406        0.2400  0.341273
```



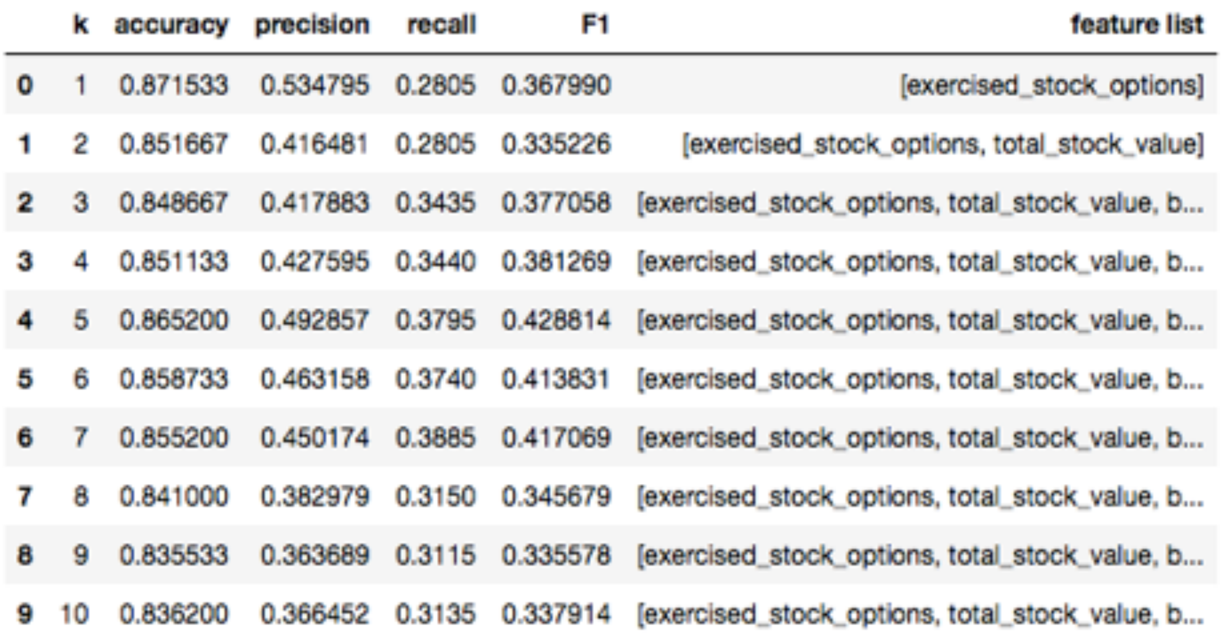
```
the scores of with new features

                     accuracy  precision_score  recall_score        F1
Naive Bayes          0.757200         0.257817        0.4370  0.324304
Decision Tree        0.795600         0.227505        0.2225  0.224975
Random Forest        0.858200         0.392190        0.1155  0.178447
Logistic Regression  0.683067         0.088710        0.1485  0.111070
LinearSVC            0.682800         0.127096        0.2350  0.164970
KNN                  0.876467         0.590406        0.2400  0.341273
```

You can notice that the performance differs from one algorithm to another.But in general,they have no too much effect on the performance while after adding the new features.

2.2 Univariate Feature Selection

In order to decide the best features to use, I use scikit-learn 'SelectKBest' to made the function which list the evaluation values(such as accuracy,precision,recall,F1) of combination with GaussianNB when K is greater than or equal to k by selected feature. I select best 10 influential features and returned the below scores in each k to see which has best score in evaluation metrics.



	k	accuracy	precision	recall	F1	feature list
0	1	0.871533	0.534795	0.2805	0.367990	[exercised_stock_options]
1	2	0.851667	0.416481	0.2805	0.335226	[exercised_stock_options, total_stock_value]
2	3	0.848667	0.417883	0.3435	0.377058	[exercised_stock_options, total_stock_value, b...
3	4	0.851133	0.427595	0.3440	0.381269	[exercised_stock_options, total_stock_value, b...
4	5	0.865200	0.492857	0.3795	0.428814	[exercised_stock_options, total_stock_value, b...
5	6	0.858733	0.463158	0.3740	0.413831	[exercised_stock_options, total_stock_value, b...
6	7	0.855200	0.450174	0.3885	0.417069	[exercised_stock_options, total_stock_value, b...
7	8	0.841000	0.382979	0.3150	0.345679	[exercised_stock_options, total_stock_value, b...
8	9	0.835533	0.363689	0.3115	0.335578	[exercised_stock_options, total_stock_value, b...
9	10	0.836200	0.366452	0.3135	0.337914	[exercised_stock_options, total_stock_value, b...

As you can see, when k is 5,F1 is the highest, and I have good precision and good recall. I decided to take top 5 features along with POI as they obtained the highest scores from results.

So my final features choice: **['exercised_stock_options','total_stock_value','bonus', 'salary', 'deferred_income']**

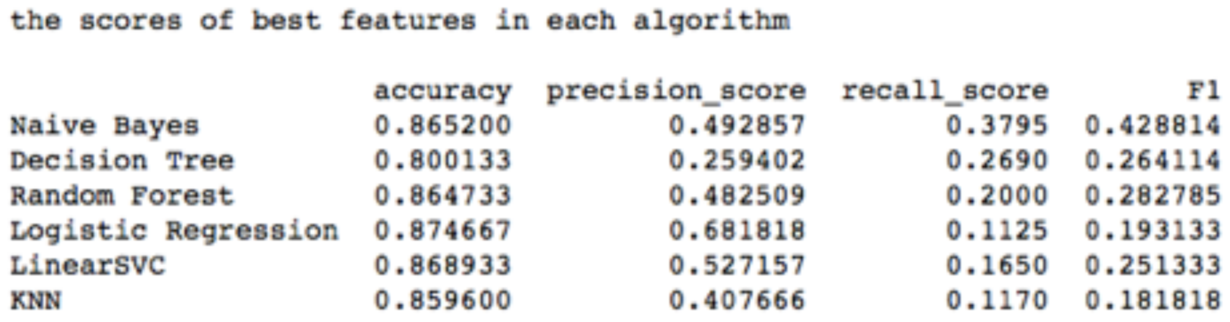
2.3 Scale feature

After feature engineering & using SelectKBest, I scaled all features using min-max scalers. Since my selected features had different units and some of the features had very big values, I needed to transform them. I used MinMaxScaler from sklearn to scale all my selected features to a given range (between 0 and 1).

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”]

- Now we are ready to train a model and predict the required solution.There are 60+ predictive modeling algorithms to choose from. Our problem is a classification and regression problem. We want to identify relationship between output (POI or not) with other variables or features (bonus,salary,...). We are also perfoming a category of machine learning which is called supervised learning as we are training our model with a given dataset. With these two criteria Supervised Learning plus Classification and Regression, we can narrow down our choice of models to a few. I tried:
    - Naive Bayes classifier
    - Decision Tree
    - Random Forrest
    - Logistic Regression
    - LinearSVC
    - KNN or k-Nearest Neighbors

- Evaluate model performance before tuning



```
the scores of best features in each algorithm

                     accuracy  precision_score  recall_score        F1
Naive Bayes          0.865200         0.492857        0.3795  0.428814
Decision Tree        0.800133         0.259402        0.2690  0.264114
Random Forest        0.864733         0.482509        0.2000  0.282785
Logistic Regression  0.874667         0.681818        0.1125  0.193133
LinearSVC            0.868933         0.527157        0.1650  0.251333
KNN                  0.859600         0.407666        0.1170  0.181818
```

The accuracy score in each algorithm is pretty high, which is an indication that accuracy in this case is not the best evaluation metric. So we look at the other respective score, I ended up choosing **Gaussian Naive Byes** which is the best performance compared to any other classifier that I tried.

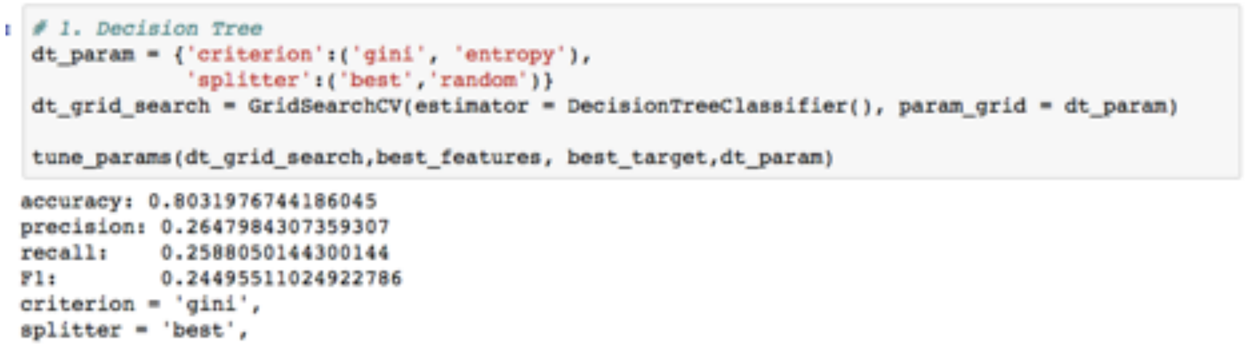
4.What does it mean to tune the parameters of an algorithm, and what can happen if you don't do this well? How did you tune the parameters of your particular algorithm? What parameters did you tune? (Some algorithms do not have parameters that you need to tune -- if this is the case for the one you picked, 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"]

Essentially, the ultimate goal of machine learning is to make a machine system that can automatically build models from data without requiring human involvement.However, before you use models,one of difficulties is that learning algorithms require you to set parameters which have been identified to affect system performance as evaluated by some appropriate metric.Thus, Tuning is the good way to select the best parameters for an algorithm to optimize its performance.And tuning in machine learning is an automated process for doing this.

To make it more concrete, Typical examples include clustering like KNN in machine learning algorithm which must specify the number neighbors.One of the ways you do can be trying many different values of the number neighbors and seeing which is the best result in your model.What I do is to use GridSearchCV from sklearn for parameter tuning which search to automatically tune the algorithm.

4.1 Tune Algorithm

- **Decision Tree**

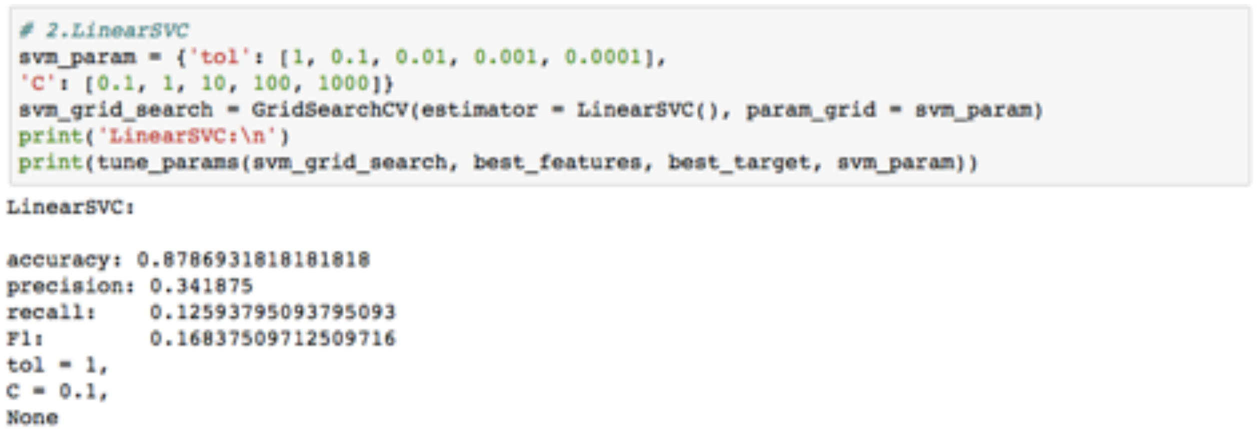


```
# 1. Decision Tree
dt_param = {'criterion':('gini', 'entropy'),
            'splitter':('best','random')}
dt_grid_search = GridSearchCV(estimator = DecisionTreeClassifier(), param_grid = dt_param)

tune_params(dt_grid_search,best_features, best_target,dt_param)
```

```
accuracy: 0.8031976744186045
precision: 0.2647984307359307
recall:    0.2588050144300144
F1:        0.24495511024922786
criterion = 'gini',
splitter = 'best',
```

- **LinearSVC**

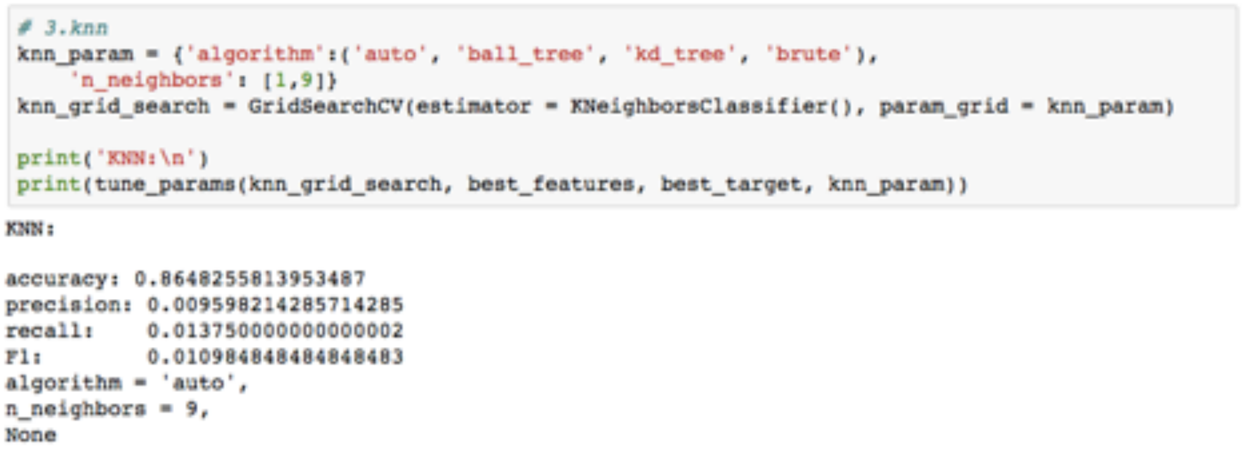


```
# 2.LinearSVC
svm_param = {'tol': [1, 0.1, 0.01, 0.001, 0.0001],
'C': [0.1, 1, 10, 100, 1000]}
svm_grid_search = GridSearchCV(estimator = LinearSVC(), param_grid = svm_param)
print('LinearSVC:\n')
print(tune_params(svm_grid_search, best_features, best_target, svm_param))
```

```
LinearSVC:

accuracy: 0.8786931818181818
precision: 0.341875
recall:    0.12593795093795093
F1:        0.16837509712509716
tol = 1,
C = 0.1,
None
```

- **KNN**



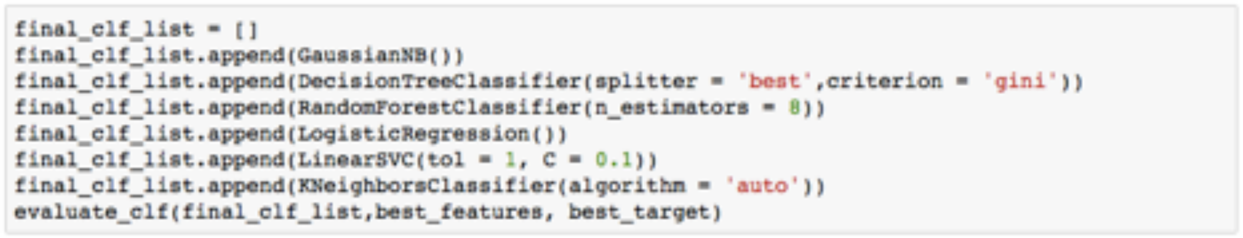
```
# 3.knn
knn_param = {'algorithm':('auto', 'ball_tree', 'kd_tree', 'brute'),
     'n_neighbors': [1,9]}
knn_grid_search = GridSearchCV(estimator = KNeighborsClassifier(), param_grid = knn_param)

print('KNN:\n')
print(tune_params(knn_grid_search, best_features, best_target, knn_param))
```

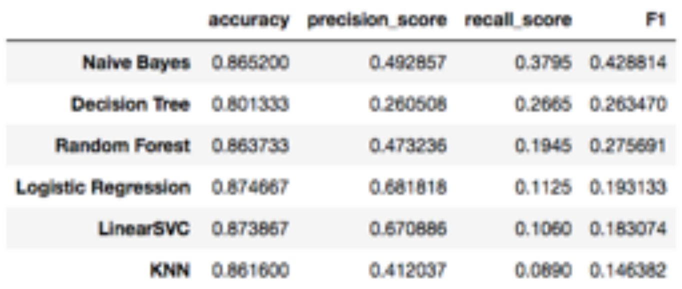
```
KNN:

accuracy: 0.8648255813953487
precision: 0.009598214285714285
recall:    0.013750000000000002
F1:        0.010984848484848483
algorithm = 'auto',
n_neighbors = 9,
None
```

4.2 Showing the evaluations in each algorithm After tuning



```
final_clf_list = []
final_clf_list.append(GaussianNB())
final_clf_list.append(DecisionTreeClassifier(splitter = 'best',criterion = 'gini'))
final_clf_list.append(RandomForestClassifier(n_estimators = 8))
final_clf_list.append(LogisticRegression())
final_clf_list.append(LinearSVC(tol = 1, C = 0.1))
final_clf_list.append(KNeighborsClassifier(algorithm = 'auto'))
evaluate_clf(final_clf_list,best_features, best_target)
```



	accuracy	precision_score	recall_score	F1
Naive Bayes	0.865200	0.492857	0.3795	0.428814
Decision Tree	0.801333	0.260508	0.2665	0.263470
Random Forest	0.863733	0.473236	0.1945	0.275691
Logistic Regression	0.874667	0.681818	0.1125	0.193133
LinearSVC	0.873867	0.670886	0.1060	0.183074
KNN	0.861600	0.412037	0.0890	0.146382

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 the process which decides the numerical results quantifying hypothesized relationship between features,are acceptable descriptions of the data.
What a classic mistake I can make is not splitting my data to training/test which leads to overfitting.

The validation does not give us an indication of how well the learner will generalize to an in dependent / unseen dataset.

Because of the small size of the dataset, StratifiedShuffleSplit validation is used.The advantage of StratifiedShuffleSplit is still keeping proportion of labels in the skew data. For example, there are ten folds, every fold will contains equal proportion of POIs vs non-POI .
If we use Cross Validation train_test_split function to split 30% of my testing data, it could be no POIs labels in the testing data ,or even worse in training data which would makes the model isn't good enough.

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 accuracy,precision,recall and F1 as my evaluation metrics.Since the accuracy score in each algorithm is pretty high, which is not the best evaluation metric.

Precision:
Proportion of all positive predictions that are correct. Precision is a measure of how many positive predictions were actual positive observations.

Recall:
Proportion of all real positive observations that are correct. Precision is a measure of how many actual positive observations were predicted correctly.

F1 Score:
F1 score is an 'average' of both precision and recall. We use the harmonic mean because it is the appropriate way to average ratios (while arithmetic mean is appropriate when it conceptually makes sense to add things up).

Although logistic regression (precison:0.68 & recall: 0.11) and LinearSVC (precison:0.67 & recall: 0.11) also have hight precision,F1 scores are lower than Gaussian Naive Byes. So I choose **Gaussian Naive Byes** as my final algorithm.

References

scikit-learn Documentation
Introduction to Machine Learning (Udacity)
https://chrisalbon.com/machine-learning/precision_recall_and_F1_scores.html