Enron poi identification

Goal of project:

The goal of project is to find the person of interest POI using finanacial data and email of the enron dataset. Here i'll be using scikit-learn to train my model.

Summary of data:

The dataset containing 146 personal and financical information of persons. We have totally 21 features other than POI all other feature having null value.

- Bonus – 64
- deferral_payment – 107
- deferred_income – 97
- email_adddress – 35
- exercised_stock_options – 44
- expenses – 51
- from_messages – 60
- from_poi_to_this_person – 60
- from_this_person_to_poi – 60
- loan_advance – 142
- long_term_incentive - 80
- other – 53
- restricted_stock – 36
- restricted_stock_deferred – 128
- salary – 51
- shared_receipt with_poi – 60
- to_messages – 60
- total_payments – 21
- total_stock_value - 20

The allocation of poi and non poi

POI: 18

NON-POI: 128

Outlier handling:

In exploratory phase I have sensed that lot of attributes are null so I counted total number of data it was 146, in that few of the attributes having null value more than 100 so that I thought the values are useless, the attributes are

- deferred_income
- director_fees
- loan_advances
- restricted_stock_deferred
- email_address
- poi

when I printed all the name I came to know that one person is not real which was `TOTAL` so I have removed that

Feature engineering:

I have created two feature additionally named are:

- to_fraction – ratio of mail recevied from poi
- from_fraction – ratio of mail sent to poi

Feature scaling:

I don’t think so feature scaling is necessay for this application because we don’t have high variant data or complex data like image binary

Features used:

for feature selection I have used selectKbest intially I have used value and I printed the score then I see that there are feature with bit higher score so I have incresed the k from 5 to 7 to select appropriate feature and here the scores are:

```
'from_fraction', 2.8709837494578281
 'salary', 3.2905618850358156
 'to_messages', 65.197357445894383
 'deferral_payments', 13.264907682062823
 'total_payments', 60.999183115442335
 'exercised_stock_options', 1.4525044063186563
 'bonus', 26.029131605787331
 'restricted_stock', 1.3598908875246158
 'shared_receipt_with_poi', 33.649854819692969
 'total_stock_value', 1.9318596601722953
 'expenses', 0.9593992930328934
 'from_messages', 4024.9076180178145
 'other', 9.7100816586398864
 'from_this_person_to_poi', 4510.4878837653941
 'long_term_incentive', 0.9813490001881896
 'from_poi_to_this_person', 14.175834805445238
```

Algorithm Used:

- LinearSVM
- Random Forest

Linear SVM showed precision performance

- percison 0.24518
- recall 0.43250
- Random Forest
- precision 0.26038
- recall 0.08650

Parameter Tuning:

we can optimize the model by tuning the parameter. It is very important to make our model to perform better because each dataset having their own nature as a machine learning engineer it’s out duty to investigate the data and to tune the model to make it perform well

I tuned the random forest by using min_samples_leaf parameter of different value such as 5,10,20 using gridseachcv then It showed better result on having value of 5

min_samples_leaf – number of minimum sample for further split

I tuned the parameter by using grid search cv. It showed high performance when it has 5 min_samples_leaf

Evaluation Metrics:

Percison score – 0.4117

Recall score – 0.4117

Precision tells the true positives to the records that are real POIs,Recall captures the ratio of true positives to the records predicted as POI