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
  the reason behind on creating from two feature is percentage of mail recived and sent from the overall activity will help us in creating the model

Feature scaling:

I have used sklearn’s scaling to scale the feature from high value to low value, in svm classification it’ll work better becasuse we scalling the value in a definite ratio, if we didn’t scale the ratio between one feature and one more feature will reduce the efficieny of the model if scaled the ratio will be normalized so svm work well

Features used:

I have tried different k value I got best prediction when k = 6 so I have used k value as 6

Algorithm Used:

- LinearSVM
- Random Forest

Linear SVM showed precision performance
- percison 0.878787878788
- recall 0.878787878788
- Random Forest
- precision 0.848484848485
- recall 0.848484848485

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

Validation :

validation is the process of validating your model, by spliting the your data into train and test. We can train our model by using train data and we can see the performace of model by using test data

Validation Strategy :

I have used holdout evaluation because the dataset is so small I don’t think so there is need of high radomization and there won’t be much high varience

Evaluation Metrics:

Percison score – 0.4117
Recall score – 0.4117

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