Fraud Detection by Machine Learning

Author : Görkem Berk ŞAHAN
Date : 17-Oct-2017

1. Summarize Project and Data Description

Aim of this project selecting the a machine learning algorithm and training it with best parameters in order to detect persons that associate with enron scandal,For this process we are staring with inspection data first. Then select some features that associated with poi we think and then cleaning,scaling (if needed) features then select an machine learning algorithm that find if person is poi according to the given features.

We have 146 rows sample data that contains a group of associated Enron scandal, 18 of them are POI(person of interest labelled) and 127 of them are Non-POI. All data row has 21 features like below,

- salary
- to_messages
- deferral_payments
- total_payments
- exercised_stock_options
- bonus
- restricted_stock
- shared_receipt_with_poi
- restricted_stock_deferred
- total_stock_value
- expenses
- loan_advances
- from_messages
- other
- from_this_person_to_poi
- poi
- director_fees
- deferred_income
- long_term_incentive
- email_address
- from_poi_to_this_person

2.Feature Selection

We are going to use some financial and email features ;

financial features: ['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'] (all units are in US dollars)

email features: ['to_messages', 'from_poi_to_this_person', 'from_messages', 'from_this_person_to_poi', 'shared_receipt_with_poi']

Outliers :

POI is not continuous , so we cant use regression model and because of size of data, we can inspect it by visual so we can find the outlier, After visual inspaction on Dataset We can see that data has an TOTAL value that is an report result. And we removed it . From Pdf file we can see that "LOCKHART EUGENE E" values Nan and "THE TRAVEL AGENCY IN THE PARK" is not valid.

Feature Selection :
we are going to use selectKbest algorithm and acorrding to the this, features scores are below, we don't know how many number of features will be best for us , we are going to test it with gridsearch.

I've added two new features (perc_from_poi and perc_to_poi) that I think that it would be valuable to find relation with POI, first one is what percent of how many mails to POI , and other one is percent of emails form POI. with this values new feature slection scores are like below ;

```
* exercised_stock_options   -> 24.8150797332
* total_stock_value         -> 24.1828986786
* bonus                     -> 20.7922520472
* deferred_income           -> 11.4584765793
* long_term_incentive       -> 9.92218601319
* restricted_stock          -> 9.21281062198
* total_payments            -> 8.77277773009
* shared_receipt_with_poi   -> 8.58942073168
* expenses                  -> 6.09417331064
* from_poi_to_this_person   -> 5.24344971337
* perc_from_poi             -> 5.12394615276
* perc_to_poi               -> 4.09465330958
* from_this_person_to_poi   -> 2.38261210823
* deferral_payments         -> 0.224611274736
* restricted_stock_deferred -> 0.0654996529099
```

3. Pick and Tune Algorithm

Our output variable is not continious, so we need a classification algorithm to detect if person is POI or not. I have tried some supervised classification algorithms like desicion tree, RandomForest, Adaboost, Support Vector Machine.

SVM Performence is better then Random forest algorithm's prformance , accuracy is close like 0.93 vs 0.9310 but when we look at presicion and recall metrics SVM is better than Random forest, so I've select SVM to use.
Random forest Recall: 0.12850
SVM Recall :0.31350

Beacuse of the data size, training time difference is not much between algorithms, if you use classification alg. with scaling features it doesnt take long time but if you dont scale features it took much more time.

Here, selected algorithms detail to **tune** and **training** steps ;

Support Vector Machine - SVM :

Using below parameter array we tried to find best combination by using gridserach ;

```
# svm
krnls = ["rbf","linear"]
c = [100000,10000,1000]
gamma = [0.001,0.01,1,10,100,1000]
```

Mising Values ;
Because of the data size, all data is valuable and we don't want to miss anything, According to a quick google search if you have less data you should keep it and to train of course outliers and much missing values must be cleared,According to the data size we can eye inspect.and I closed "Imputer", after closing it, presicion and recall values increased.

Fitting results of gridsearch ;

```
gridserach time : 125.84 s
best params : {'clf__gamma': 1, 'pca__n_components': 4, 'selection__k':
9, 'clf__C': 100000, 'clf__kernel': 'rbf'}
```

Validation :

I've holded out 20% of data to test the trained data and result is

train time : 0.16 s
score clf : 0.931034482759

Tester results :

```
Pipeline([
      ('scaler', MinMaxScaler()),
      ('selection', SelectKBest(k=9)),
      ('pca', PCA(n_components=4)),
      ('clf', SVC(C=100000,kernel="rbf",gamma=1))
 ])

Accuracy: 0.81733       Precision: 0.31444      Recall: 0.31350 F1: 0.3
1397     F2: 0.31369
Total predictions: 15000        True positives:  627    False positives
: 1367   False negatives: 1373   True negatives: 11633
```

4. Tunning Parameters

Selection of Parameter list is important I think, if you gave wrong combinations to grid , it gave you the best parameter combination but this may not help you to find true answers in limitation that you expect. Fo r example it find 1% of true output but this doesn't help you.

Aim of the performance tunning , making your algorithm to provide true output upper or equal limitation that you decide, Generally limitation is upper is better because your it means that algorithm gives true output everytime.

After gridsearch fitting, I consider test results and re think about grid parameters for example should I add new C values to my list and execute again grid with new ones.

5. Validation :

The common mistake is training model with hole data without hold out test data, it causes overfitting it and Accuracy number would be so high. We should check which data used in training set and wich is in testing set, by using crossvalidation and random_state number we can provide it. I've holded out 20% of data to test the trained data and result random_State number is 42.

6. Evaluation Metrics :

We will use two metrics that Presicion and Recall values ; Presicion means that ratio of true positive output between all positive output findings, if your algorithm finds all positive outputs and all of them is really positive then presicion is high and equals to 1. At the same time Recall means that true positive ratio of all positive labelled values.

I've create an custom scoring function that calculating presicion and reacall and returning minimum value of them, and setted greater is better parameter to true to choose best params on gridseach.By this way if I will consider the lower one of my algorithms presicion or recall values so I would scored it according to the weakest ring.

Referances :

http://scikit-learn.org (http://scikit-learn.org) , SVM, DescionTree, Make_scorer, Adaboost, RandomForest
http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.StratifiedShuffleSplit.html (http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.StratifiedShuffleSplit.html)
https://github.com/rhiever/tpot/issues/301 (https://github.com/rhiever/tpot/issues/301)
https://chrisalbon.com/machine-learning/svc_parameters_using_rbf_kernel.html (https://chrisalbon.com/machine-learning/svc_parameters_using_rbf_kernel.html)
Udacity Data analyst - Intro the machine learning lessons
Udacity Disccassion Forums

In [ ]: