# Person of interest identifier(write-up)

1). The goal of this project is to train an algorithm that effectively identifies a person of interest in the Enron fraud. To state clearly a person of interest, in this case, is any person that has been involved in any sort of fraudulent transactions, for instance, stealing money from the company, taking part in fraudulent transactions etc.

2) In my POI identifier, I ended up using selectkbest to select features. I used the standard scaler to scale my features since it was a requirement for some estimators eg gaussian( which was used).Since some of the different features in the dataset had different ranges the dataset had to be scaled to make data more digestible to the algorithms.

Also, I tried to engineer a feature that gave the ratio between the total stock value and their salary, how did I come up with this? well, i figured the poi my must be securing money for themselves in some form or fashion and my best guess was that it would be stocked so I computed the ratio between their total stock value and their payment for work(Salary) to see how much more their total stock value is more than their salary.

Feature p values:  [  5.75289462e-03   7.90100377e-01   8.97764213e-03   7.17333771e-01

  3.66342015e-07   1.37546522e-04   9.07062076e-06   4.33113589e-03

  2.59641887e-04   1.29626320e-01]

Feature scores:  [  7.88322914   0.07114218   7.03569408   0.13162892  28.71240726

 15.43407565  21.3422848    8.43063369  14.10018684   2.32615615]

Furthermore, the parameters in the classifiers were carefully tuned to ensure that they generalize well. Therefore, in this poi identifier the parameters class weight, max depth, and random seed were tuned.

1. Since the dataset is imbalanced, I thought the minor class was probably being ignored. So to place more emphasis on the minor class, the classifiers which had the class weight parameter, for instance, Logistic regression, SGDClassifier and Random Forest, were tuned make sure the classifier puts more emphasis on the minor class. Which I believe kept the classifier from underfitting
2. Also, the max depth in the decision tree based classifier was set to very low values to avoid overfitting
3. The random state parameter was tuned to ensure that there is reproducibility in the results

3) Additionally, the classifier I used in my person of interest identifier were Random forests, SGDClassifier, Gaussian NB and a logistic classifier. These classifiers alone produce fairly good results but on the evaluation set the results were minimal, so I used an ensemble voting classifier utilizing all the classifiers that were trained and ultimately producing better one.

4) Furthermore, tuning hyperparameter is the process of choosing optimal parameters for an algorithm that will optimize its ability to generalize well. If this is done well the algorithm won't miss the relevant relationship between input features and target output nor will it become overly complex. In this project, the parameters in the classifiers were carefully tuned to ensure that they generalize well. In this poi identifier the parameters class weight, max depth, and random seed were tuned.

1. Since the dataset is imbalanced, the minor class was being ignored. So to place more emphasis on the minor class, the classifiers which had the class weight parameter, for instance, Logistic regression, SGDClassifier and Random Forest, were tuned make sure the classifier puts more emphasis on the minor class. Which I believe kept the classifier from underfitting
2. Also, the max depth in the decision tree based classifier was set to very low values to avoid overfitting
3. The random state parameter was tuned to ensure that there is reproducibility in the results

5) validation is the process of checking your model to see if appropriately approximate that of the real world by train your algorithm and testing it on the validation set the algorithm hasn't seen before. One common mistake that can be made whilst doing this is training and validating your algorithm on the same data, which will give positive feedback.

6) The evaluation metrics used were the recall, precision, and accuracy score.

GaussianNB

The accuracy is; 0.925925925926

The precision is: 0.8

the recall is  0.571428571429

Confusion Matrix

[[46  1]

[ 3  4]]

LogisticRegression

The accuracy is; 0.925925925926

The precision is: 0.8

the recall is  0.571428571429

Confusion Matrix

[[46  1]

[ 3  4]]

SVM

The accuracy is; 0.796296296296

The precision is: 0.8

the recall is  0.571428571429

Confusion Matrix

[[39  8]

[ 3  4]]

Decision Tree Classifier

The accuracy is; 0.888888888889

The precision is: 0.6

the recall is  0.428571428571

Confusion Matrix

[[45  2]

[ 4  3]]

SGDClassifier

The accuracy is; 0.888888888889

The precision is: 0.555555555556

the recall is  0.714285714286

Confusion Matrix

[[43  4]

[ 2  5]]

Ensemble(Voting Classifier)

The accuracy is; 0.944444444444

The precision is: 1.0

the recall is  0.571428571429

Confusion Matrix

[[47  0]

[ 3  4]]

The accuracy

Since the dataset is imbalanced (the number of non-poi greatly outnumber the number of poi), using accuracy as an evaluation metric can be quite misleading. This is because the classifier can easily under fit yet give seemingly good results. What happens is the classifier classifies every data point as the major class (non-poi) ignoring the minor class.

Confusion matrix

This is a more holistic evaluation metric than accuracy since it gives an illustration of how confused the classifier is. It returns a matrix wherein the 1st row contains the true negatives which are the number non-poi predicted correctly and next is the false positive which is the number of non-poi the classifier confused as poi. The 2nd row contains the false negatives the number of poi the classifier confused as non-poi and the true positives the number of persons the classifier predicted correctly as poi.