Project 5 Identify Fraud from Enron Email

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"]**

   The dataset we use in this project is constructed by the email and financial data from senior manager of Enron. Some of them are related to the Enron Company Fraud.

   The goal of this project is to find the PIO (Person of Interest) from the whole group. Features are extracted from both the email and financial data, and we can also construct new feature according to our understanding.

   Some of the features are evaluated and selected. Different classifiers with verified algorithm will be built based on the selected features. The classifiers are used to label the each observation as PIO or not PIO. We will evaluate the performance of our classifier by precision, recall and F1 score.

   For the data set, there are 146 data points. Among them, 18 points are poi, and 128 are not poi points. For each data point, it include 22 feature dimensions including the "poi".

   For each feature dimensions, the "poi" feature don't have any missing value "Nan". But other features all possess some missing value. For example, the feature "loan_advances" has as much as 142 missing value, and 128 for "restricted_stock_deferred".

   During my exploration, I find the outlier "TOTAL" through visualization. Then I popped it out from the dataset.

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", "properly scale features", "intelligently select feature"]

Finally, i choosed these 4 features.

- salary
- exercised_stock_options
- bonus
- total_stock_value

At first, I delete the „email" from the feature list. Because it is just the email addresses of the observations. That does not show any information for identification process.

I used *grid_search* method to find the best parameter for my kBest selector.

The best parameter turns out to be *k=4.*

Then I feed the features data into the kBest selector again, to find the score of each feature. They are shown below:

salary	18.57570327
exercised_stock_options	25.09754153
bonus	21.06000171
total_stock_value	24.46765405
deferred_income	11.59554766
expenses	6.23420114
total_payments	8.86672154
long_term_incentive	10.07245453
restricted_stock	9.34670079
to_messages	1.69882435
from_poi_to_this_person	5.34494152
from_messages	0.1641645
from_this_person_to_poi	2.42650813
shared_receipt_with_poi	8.74648553
stock_to_salary_ratio	0.11604758

I did not use feature scaling. Firstly, when I used scaling, the performance of my classifier didn't get obviously better. Secondly, the unit of selected features are all USD dollars. It is meaningful unit in the real world. So the disparity may indicate the difference between the observations.

I created feature "stock_to_salary_ratio", because I didn't know that, whether there is a relationship between the stock value and salary of each employee. I thought, maybe the Stock value is of more importance for POIs rather than non-POIs. If this assumption stands, POIs generally have a higher ratio than the non-POIs. But it turns out to be not true. In terms of metrics, this feature is not a successful one. It get only 0.1 for the feature score. So it is not selected for the training process.

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"]**

	Accuracy	Precision	Recall	F1	parameters
Naive Bayes	0.85053	0.42254	0.33000	0.37058	{'fselect__k': 4}
KNeighbors	0.87800	0.69602	0.36750	0.48102	{'knn__algorithm': 'brute', 'knn__leaf_size': 1, 'knn__n_neighbors': 3, 'knn__weights': 'distance'}
Decision tree	0.83000	0.34481	0.30550	0.32397	{'tree__criterion': 'gini', 'tree__max_features': 2, 'tree__min_samples_split': 3}

I picked Kneighbors, because it has the best performance, no matter in which indicator.

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? (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 item: "tune the algorithm"]

The parameters deeply influence the behavior and performance of an algorithm. Generally speaking, we tune the algorithm to increase the accuracy of our model.

For example, when the k value of k nearest neighbor algorithm is tuned, one observation may be classified as different labels, which lead to a higher or lower accuracy and recall.

Another concerning for tuning is the kernel trick. A kernel trick is actually a mapping function, which transform the data points from the origin space into a new space, where they can be easily separated.

For parameter gamma, it can be understood as the influence radius from each data point on the model.

For parameter C, it is the tradeoff between model simplicity and misclassification of training examples. A low value for C will result into smooth classification boundary between classes, which also lead to more misclassification.

Overfitting is another problem. If the c value of support vector machine is tuned badly, it may result into overfitting. This will deeply harm its performance on any test dataset other than the training data. And it will increase the computational complexity of our algorithm，and performance increase for specific data set at the cost of performance on any other data set.

When we use select k best algorithm to evaluate the features, we may lose some information if we set a too low k value, and we may increase some noise to our classifier if we set a too high k value.

I used pipeline and GridSearchCV in the sklearn to find the best parameter of my algorithm, in order to optimize the F1 score.

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 item: "validation strategy"]**

Validation is the test for the performance of our model. During the validation, our model is tested with data that different from the training data. The predicted result should be compared with the test dataset labels. The main indicators include accuracy, precision, recall and f1 score.

The classic mistake include that, we may sample from the origin data in certain sequence, but not randomly. This will result into extremely high or low performance of our model. Because the test data can't show the general trend or pattern of the whole population.

I use the function StratifiedShuffleSplit to separate the training and test data. In order to find a better evaluation, I made 100 iteration to avoid fortuity.

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"]**

	Accuracy	Precision	Recall	F1
KNeighbors	0.87800	0.69602	0.36750	0.48102

Here is my evaluation metrics table.

Accuracy indicates that, how likely is the model right, no matter it identify a given observation as negative or positive.

Precision indicate that, when our model identify an observation as positive, how likely should I believe it. For our example here, our model has 0.696 as its precision. Then if the model identifies N observations as PIOs, We can say that only 0.696*N observations are true POIs.

Recall indicates that, given certain number of positive observations, how well could this model identify them as positive. For our example here, our model has 0.36750 as its recall. Then if N PIOs exists in the dataset, We can say that only 0.3675*N observations can be identified out as POI by our model.

F1 is the combination of precision and recall, in other words, a balance between them.

Generally speaking, my model has a well accuracy, but it is a little weak for identify the true positive from all positive observations.