Introduction

The Enron scandal, publicized in October 2001, eventually led to the bankruptcy of the Enron Corporation, an American energy company based in Houston, Texas, and the de facto dissolution of Arthur Andersen, which was one of the five largest audit and accountancy partnerships in the world. In addition to being the largest bankruptcy reorganization in American history at that time, Enron was cited as the biggest audit failure. [1]

In the resulting Federal investigation, a significant amount of typically confidential information entered into the public record, including tens of thousands of emails and detailed financial data for top executives. [2]

Who are the persons of interest in the fraud case? Based financial data available, we are going to build an identifier to find out the persons of interest.

Data analysis

The data analyze has 146 observations. The numbers of Persons of interest (POI) identified is 18 and the allocation across classes is 0.14. The total number of feature available is 20.

Missing value

In real world, handling missing value is one of the greatest challenge faced by analyst, because making the right decision on how to handle it generates robust data models. [3]

The next table show the proportion of missing value by feature.

Feature	Number	Percent of missing value
salary	51	34.93 %
to_messages	60	41.10 %
deferral_payments	107	73.29 %
total_payments	21	14.38 %
exercised_stock_options	44	30.14 %
bonus	64	43.84 %
restricted_stock	36	24.66 %
restricted_stock_deferred	128	87.67 %
total_stock_value	20	13.70 %
director_fees	129	88.36 %
from_poi_to_this_person	60	41.10 %
loan_advances	142	97.26 %
from_messages	60	41.10 %
other	53	36.30 %
expenses	51	34.93 %
from_this_person_to_poi	60	41.10 %

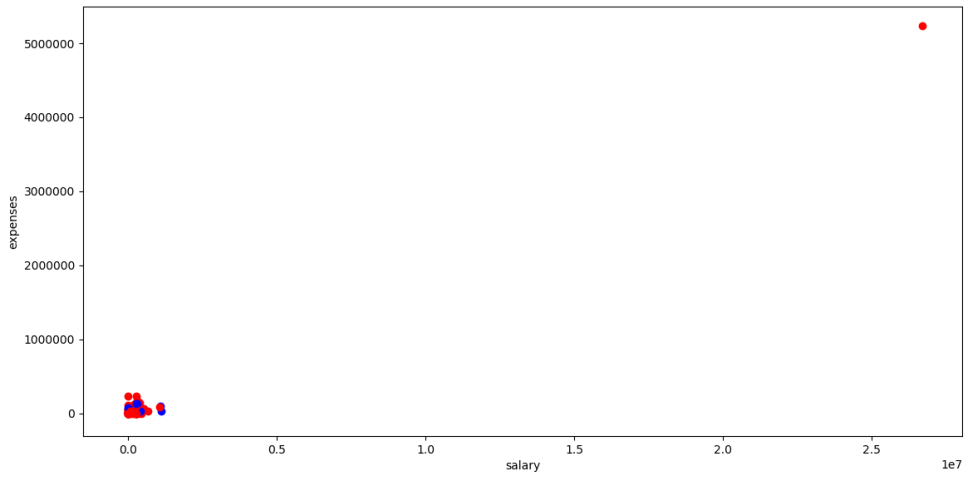
deferred_income	97	66.44 %
shared_receipt_with_poi	60	41.10 %
email_address	35	23.97 %
long_term_incentive	80	54.79 %

The last table showed there are features with more than 50% of missing value.

I think whether we consider feature with more than 50% of missing value we can make mistake to find out the person of interest because the dataset has few observations and we are missing half of the dataset.

Outlier Investigation

An outlier is an observation that lies an abnormal distance from other values in a random sample from a population. In a sense, this definition leaves it up to the analyst (or a consensus process) to decide what will be considered abnormal. Before abnormal observations can be singled out, it is necessary to characterize normal observations. [4]



After plot the salary vs expense, I figured out an outlier value.
A commonly used rule says that a data point is an outlier if it is more than 1.5 * IQR above the third quartile or below the first quartile.[6]

IQR = Q3 – Q1, where Q3 = median of the n largest entries and Q1 = median of the n smallest entries.[7]

I count how many outliers there were for each data entry and print the top 14.

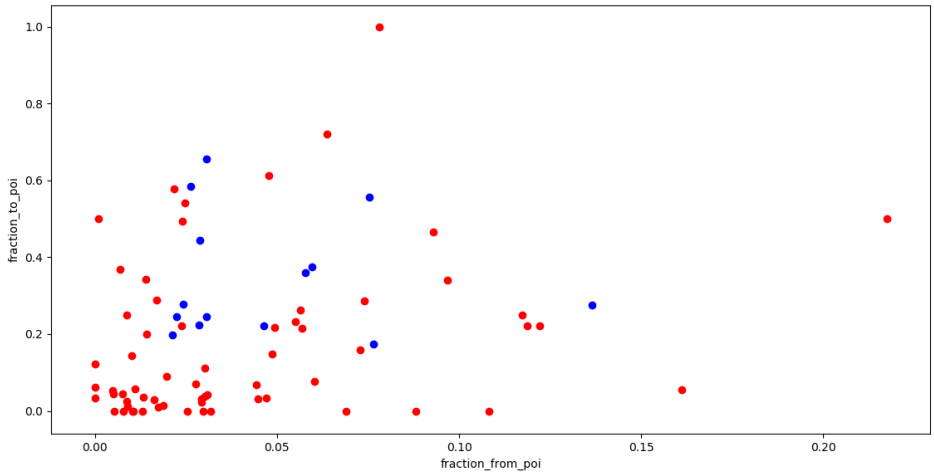
Name	Number of outliers
TOTAL	14
FREVERT MARK A	10
BELDEN TIMOTHY N	10
LAY KENNETH L	10
WHALLEY LAWRENCE G	8
LAVORATO JOHN J	8
SKILLING JEFFREY K	8

ALLEN PHILLIP K	6
BAXTER JOHN C	6
RICE KENNETH D	6
KITCHEN LOUISE	6
DELAINEY DAVID W	6
DERRICK JR. JAMES V	6
SHAPIRO RICHARD S	5

The first one is the TOTAL. According the official pdf documentation [8], Total is the sum of the columns and we can drop it from the dataset.

However, there are 3 person with 10 outlier feature. They are FREVERT MARK A, BELDEN TIMOTHY N and LAY KENNETH L. According the dataset, FREVERT MARK A is not a POI, and BELDEN TIMOTHY N and LAY KENNETH L both are. I don't think we should remove them from the dataset because they are important for our analyses.

Next plot is going to show the salary vs expense without the TOTAL row.



There is payments were made by Enron employees of business-related travel to 'THE TRAVEL AGENCY IN THE PARK' [8]. The data in the dataset is about Enrom employees and this company will be dropped from the dataset.

Features selection.

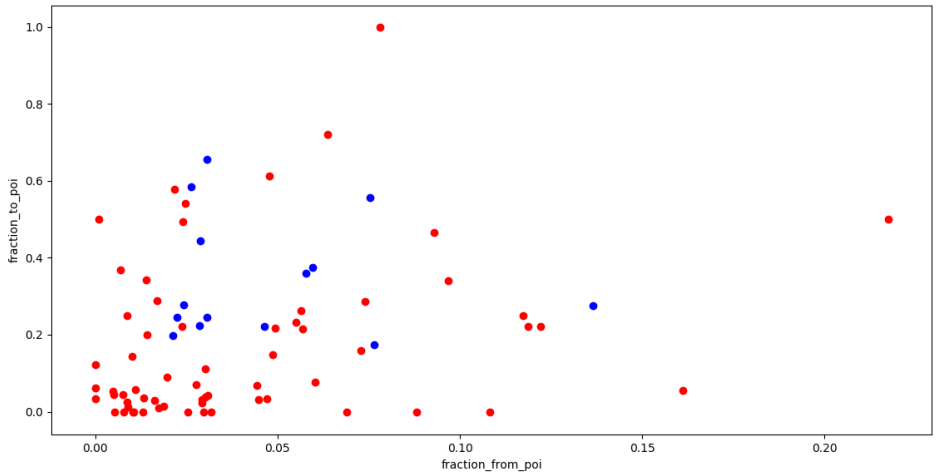
Feature selection is the process of selecting a subset of relevant features (variables, predictors) for use in model construction. Feature selection techniques are used for four reasons: simplification of models to make them easier to interpret, shorter training times, to avoid the curse of dimensionality, enhanced generalization by reducing overfitting. [5]

I am putting away the features, which there are missing value higher than 50%. The features we are going to work with are: salary, expenses, total_payments,

exercised_stock_options, bonus, restricted_stock, total_stock_value, 'from_poi_to_this_person, shared_receipt_with_poi, from_messages.

New feature.

I create two new features named fraction from poi and fraction to poi.
Fraction to poi is the number emails sent to a POI divided by the total of email sent.
Fraction from poi is the number of emails received from a POI divided by the total of e-mail received.



Based on the plot, the fraction_to_poi lower than 0.2 is from no POI person. Maybe this is a good feature to be used for our classifier, so I'm going to include this new features in my list of features.

Feature scaling.

The feature we are using don't have the same scale. A clear example is the salary and the total stock value. The salary is a fixed amount of money paid every year and the total stock value can be the savings of a life, what means it can be a much higher value. In the dataset, the HANNON KEVIN P prove it. His salary is 243,293.00 and the total stock value is 6,391,065.00.

Metrics.

In order to reduce the number of features we are going to use 4 metrics: accuracy, precision, recall and f1-score.

Accuracy is the number of items in a class labeled correctly divided by all the items in the class[2].

Precision is the number of true positive divided by the number of true positive plus false positive. (True positive / true positive + false positive) [2]

Recall is the number of true positive divided by the number of true positive plus false negative. (True positive / true positive + false negative) [2]

F1-score can be interpreted as a weighted average of the precision and recall, where an F1 score reaches its best value at 1 and worst score at 0. F1 = 2 * (precision * recall) / (precision + recall)[10]

For our investigation, accuracy is not a good metrics because we have few POI and many is not. For this case, the likelihood it find out a no POI is very high.

Models

Gaussain Naïve Bayes, Decision Tree, SMV (SVC) and KNN are the classifier models we are going to work.

We will reduce the number of features using SelectKBest and PCA and analyze the metrics that best predict the POI.

Training and testing.

The data will be splitted in training and testing data. In order to test, we will work with 40% of the data and 60% to train.

Tunning the models.

GridSerchCV is an exhaustive search over specified parameter values for an estimator [11].

In order to get the best result from the models, we will test different parameter for each model and use the GridSerchCV to find the best one.

Decision Tree parameters:

min_sample_split: 2, 3 and 4

criterion: gini and entropy

SVC parameters:

Kernel: linear and rbf.

C: 0.001, 0.01, 0.1, 1 and 10.

Gamma: 0.001, 0.01, 0.1 and 1

KNN parameters:

n_neighbors: 1, 3, 5, 7, 10

All features.

The next table shows the prediction metrics values for all features.

	Accuracy	precision	recall	f1-score
GaussianNB	0.84	0.83	0.84	0.84
Decision_tree	0.86	0.86	0.86	0.86
SVC	0.89	0.80	0.90	0.85
KNN	0.89	0.80	0.90	0.85

SelectKBest

SelectKBest selects the features according to the k highest score. [9] We are going to use it to reduce the number of features and analyze the prediction metrics values after reducing. The K numbers of features we are going to use are 2, 4, 7 and 9.

2 features

	Accuracy	precision	recall	f1-score
GaussianNB	0.86	0.92	0.86	0.88
Decision_tree	0.86	0.88	0.86	0.87
SVC	0.91	0.91	0.91	0.91
KNN	0.87	0.85	0.88	0.86

4 features

	Accuracy	precision	recall	f1-score
GaussianNB	0.81	0.93	0.81	0.84
Decision_tree	0.84	0.80	0.84	0.82
SVC	0.87	0.80	0.88	0.84
KNN	0.79	0.82	0.79	0.81

7 features

	Accuracy	precision	recall	f1-score
GaussianNB	0.77	0.93	0.78	0.82
Decision_tree	0.84	0.88	0.84	0.86
SVC	0.89	0.80	0.90	0.85
KNN	0.89	0.80	0.90	0.85

9 features

	Accuracy	precision	recall	f1-score
GaussianNB	0.75	0.90	0.76	0.80
Decision_tree	0.79	0.86	0.79	0.82
SVC	0.89	0.80	0.90	0.85
KNN	0.89	0.80	0.90	0.85

PCA – Principal component analysis

PCA is linear dimensionality reduction using Singular Value Decomposition of the data to project it to a lower dimensional space [12].

The next tables are going to show the metrics values after reduce the number of features using the PCA statistical procedure.

2 components

	Accuracy	precision	recall	f1-score
GaussianNB	0.89	0.88	0.90	0.89
Decision_tree	0.82	0.85	0.83	0.84
SVC	0.87	0.80	0.88	0.84
KNN	0.89	0.87	0.90	0.87

4 components

	Accuracy	precision	recall	f1-score
GaussianNB	0.84	0.83	0.84	0.84
Decision_tree	0.81	0.80	0.81	0.80
SVC	0.89	0.80	0.90	0.85
KNN	0.89	0.80	0.90	0.85

7 components

	Accuracy	precision	recall	f1-score
GaussianNB	0.86	0.86	0.86	0.86
Decision_tree	0.81	0.80	0.81	0.80
SVC	0.89	0.80	0.90	0.85
KNN	0.89	0.80	0.90	0.85

9 components

	Accuracy	precision	recall	f1-score
GaussianNB	0.84	0.86	0.84	0.85
Decision_tree	0.84	0.83	0.84	0.84
SVC	0.89	0.80	0.90	0.85
KNN	0.89	0.80	0.90	0.85

Summary

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”]

My goal in this project is to build an identifier to find out the persons of interest. The dataset has 146 observations and the numbers of Persons of interest identified is 20, which means the allocation across classes is 0.14. There were many outliers however just one outlier, TOTAL row, was removed from the dataset due to the few number of data in 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”, “intelligently select features”, “properly scale features”]

There are 20 features in the dataset. I put away the features with missing value features higher than 50%, what reduce the amount of features to 10 and then I used the selectkbest function and the number of features felt down to two.

The follow tables prove that 2 features has a better performance to identify the POI than 10 features.

10 features:

	Accuracy	precision	recall	f1-score
GaussianNB	0.84	0.83	0.84	0.84
Decision_tree	0.86	0.86	0.86	0.86
SVC	**0.89**	**0.80**	**0.90**	**0.85**
KNN	0.89	0.80	0.90	0.85

2 features:

	Accuracy	precision	recall	f1-score
GaussianNB	0.86	0.92	0.86	0.88
Decision_tree	0.86	0.88	0.86	0.87
SVC	**0.91**	**0.91**	**0.91**	**0.91**
KNN	0.87	0.85	0.88	0.86

Before this analysis, I had to do scaling because the features didn't have the same scale. For example, in the dataset, the HANNON KEVIN P have salary of 243,293.00 and the total stock value is 6,391,065.00.

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"]

I end up using the SVC algorithm and I tried GaussianNB, Decision Tree and KNN with different numbers of features and parameters for each algorithm. The algorithms performance was very close, the worst f1-score was 0.86 and the best was 0.91.

Follow the table with the performance for each algorithm.

	Accuracy	precision	recall	f1-score
GaussianNB	0.86	0.92	0.86	0.88
Decision_tree	0.86	0.88	0.86	0.87
SVC	**0.91**	**0.91**	**0.91**	**0.91**
KNN	0.87	0.85	0.88	0.86

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? What parameters did you tune? (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 items: "discuss parameter tuning", "tune the algorithm"]

An algorithm can have many parameters with default value. When we don't change any parameter, it can have a worse performance than when you use the correct parameter. In order to experiment different parameters, I used the GridSearchCV automation for decision tree, SVC and KNN algorithm. SVC had the best performance and I tried different values for the parameter Kernel (Linear, RBF), C (0.001, 0.01, 0.1, 1, 10) and Gama (0.001, 0.01, 0.1, 1, 10).

The next table shows the SVC tunned had a better performance than the SVC with default parameter values.

	Accuracy	precision	recall	f1-score
SVC Tunned	**0.91**	**0.91**	**0.91**	**0.91**
SVC	0.89	0.80	0.90	0.85

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 items: "discuss validation", "validation strategy"]

The main mistake we can make is to use the same data for training and testing because you will go overfitting the data. Your algorithm is going to have bad performance when you train the data with a data class and test it with another.

In my case, I use 40% for testing and 60% for training. There is a better way to train your data if I used the k-fold cross-validation. In the k-fold cross-validation, the original sample is randomly partitioned into k equal sized subsamples. Of the k subsamples, a single subsample is retained as the validation data for testing the model, and the remaining k − 1 subsamples are used as training data. The cross-validation process is then repeated k times (the folds), with each of the k subsamples used exactly once as the validation data. The k results from the folds can then be averaged to produce a single estimation. The advantage of this method over repeated random sub-sampling is that all observations are used for both training and validation, and each observation is used for validation exactly once[13].

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"]

The 2 evaluation metrics I used were the precision and the recall. Precision is the fraction of the POI identification that are relevant for the query. Recall is the fraction of the relevant POI successfully identified.

I didn't use accuracy because we have few person of interest and many is not. For this case, the accuracy to find out a no POI is very high.

The next table show the precision and recall average for each algorithm.

	precision	recall
GaussianNB	0.92	0.86

Decision_tree	0.88	0.86
SVC	**0.91**	**0.91**
KNN	0.85	0.88

Reflection

I learned a lot during the project about pipeline, GridSearchCV, SelectKBest and PCA algorithm. When I was trying to implement the GridSearchCV and pipeline I read many things on Internet, even though I struggled to know how to work using these algorithms.

I think it was very easy to split the data, fit and predict. All the algorithms I worked had the same pattern, however, the process to tune the algorithms was very complicated. There is few information on Internet about the meaning of each parameter. Then I tried some range of values to try to find the best one.

I was surprised two features is enough to my model identify the person of interest. The data is very close to the real data. There are many missing information and the decision about the features we should use is very close to real work. I think it was amazing to work in this project and I learned a lot.

References

[1] - https://en.wikipedia.org/wiki/Enron_scandal

[2] - https://classroom.udacity.com/nanodegrees/nd115/

[3] - https://analyticsindiamag.com/5-ways-handle-missing-values-machine-learning-datasets/

[4] - https://www.itl.nist.gov/div8988/handbook/prc/section1/prc16.htm

[5] - https://en.wikipedia.org/wiki/Feature_selection

[6] - https://www.khanacademy.org/math/statistics-probability/summarizing-quantitative-data/box-whisker-plots/a/identifying-outliers-iqr-rule

[7] - https://en.wikipedia.org/wiki/Interquartile_range

[8] - https://github.com/udacity/ud120-projects/blob/master/final_project/enron61702insiderpay.pdf

[9] - http://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectKBest.html#sklearn.feature_selection.SelectKBest

[10] - http://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html

[11] - http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

[12] - http://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html

[13] - https://en.wikipedia.org/wiki/Cross-validation_(statistics)#k-fold_cross-validation