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 energy company Enron enacted bankruptcy after executing several accounting and tax frauds. The purpose of this project is to use data analysis and machine learning features like features selection, preprossessing and evaluating models to tell if an employee has been involved with fraud characterizing themselves as a person of interest (POI).

The data set used is composed of financial information, the emails sent and received by the people of interest (POI) and the identifier of whether the key person is a POI. Jeffrey Skilling (CEO) and Kenneth Lay (chairman and CEO) have many above average values, but justified by the importance of their positions.

The dataset attributes such as salary, bonuses, actions, incoming and outgoing e-mail numbers were investigated and cleaned up.

When I was looking for outliers I found a POI called TOTAL.

Does seen quite right, we have a outlier called TOTAL. This doesn't looks like a person name, in fact is the sum of all values. The second largest value receve was Kenneth Lay, but, as you know this make perfect sense since he was the CEO and chairman of Enron. We gonna delete TOTAL row.

The record "THE TRAVEL AGENCY IN THE PARK" has been deleted. Although "THE TRAVEL AGENCY IN THE PARK" had received $ 350,000 in payments 2 days before Enron's bankruptcy and Sharon Lay (sister of Kenneth Lay) owned 50% of the company, this record will not be considered a POI.

I found others persons in the dataset that need our analysis.

First of all "LOCKHART EUGENE E" doesn't have any value assign. I do not need to keep it in dataset. "CHAN RONNIE" and "POWERS WILLIAM" had stock or income put off to a later time causing a total payments equal a 0. Differently of "POWERS WILLIAM", "CHAN RONNIE" had no message sent or received, his presence on the data set is not justified. POWERS WILLIAM was save cause he have emails values. None of them are POI.

1. 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”]

After clearing the data I created two new features ['message\_poi\_ratio', 'message\_others\_ratio'] one with the percentage of emails related to the POI, both sent and received and another with the unrelated message sum. With this change I deleted all these variables 'email\_address', 'from\_messages', 'from\_poi\_to\_this\_person', 'from\_this\_person\_to\_poi', 'shared\_receipt\_with\_poi', 'to\_messages'.

With the dataset created the first thing I did was to apply the pandas.DataFrame's corr function on all variables to get the correlation between them.

With an initial idea of the data, I chose not to use the variables [total\_payments', 'total\_stock\_value'] because they are the sum of other variables in the dataset.

To get to a good number of variables to be used I used the SelectKBest function with:

1) Anova stats on a raw version of the data

2) Chi-2 stats with a standarize (MinMaxScaler) version of data

|  |  |  |  |
| --- | --- | --- | --- |
| Index | Feature | Anova | Chi-2 |
| 0 | bonus | 36.778.900 | 7.638.184 |
| 12 | message\_poi\_ratio | 16.448.332 | 5.680.295 |
| 11 | salary | 16.279.541 | 2.923.780 |
| 4 | exercised\_stock\_options | 2.236.481 | 7.836.464 |
| 6 | loan\_advances | 6.954.889 | 6.369.138 |
| 7 | long\_term\_incentive | 6.468.488 | 1.797.035 |
| 2 | deferred\_income | 6.298.692 | 0.161755 |
| 5 | expenses | 5.562.326 | 1.252.170 |
| 9 | restricted\_stock | 4.920.009 | 1.721.534 |
| 8 | other | 2.716.646 | 1.243.399 |
| 3 | director\_fees | 1.767.123 | 1.402.115 |
| 10 | restricted\_stock\_deferred | 1.138.212 | 0.037532 |
| 13 | message\_others\_ratio | 0.198870 | 0.065178 |
| 1 | deferral\_payments | 0.001801 | 0.000841 |

1. 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”]

The answer I was pursuing was to know, based on my data, whether a random person was a POI or not. This is a classification task and to help me define the way forward, I have tested multiple classifiers and their default parameters.

|  |  |  |  |
| --- | --- | --- | --- |
| Model | Precision | Recall | Accuracy |
| KNeighbors | 0.581633 | 0.0570 | 0.868800 |
| RandomForest | 0.460692 | 0.1465 | 0.863333 |
| ExtraTreesClassifier | 0.423295 | 0.1490 | 0.859467 |
| AdaBoost | 0.418319 | 0.3060 | 0.850733 |
| NearestCentroid | 0.376596 | 0.2655 | 0.843467 |
| GradientBoostingClassifier | 0.263557 | 0.2260 | 0.812600 |
| DecisionTree | 0.257143 | 0.2475 | 0.804333 |
| LogisticRegression | 0.164218 | 0.1900 | 0.763067 |
| Naive Bayes | 0.219135 | 0.6230 | 0.653733 |

Based on the accuracy, precision and recall value, I chose AdaBoost and NearestCentroid to improve performance. My goal was to achieve a precision and real value higher than 0.3, in this case, with AdaBoost I got this result even with default parameters.

Values obtained after tuning:

|  |  |  |  |
| --- | --- | --- | --- |
| Model | Precision | Recall | Accuracy |
| KNeighbors | 0.39456 | 0.37700 | 0.83980 |
| RandomForest | 0.43103 | 0.13750 | 0.86080 |
| ExtraTreesClassifier | 0.22082 | 0.68300 | 0.63640 |
| AdaBoost | 0.36377 | 0.63950 | 0.80280 |
| NearestCentroid | 0.35733 | 0.61550 | 0.80113 |
| GradientBoostingClassifier | 0.31915 | 0.23250 | 0.83153 |
| DecisionTree | 0.26790 | 0.75200 | 0.69293 |
| LogisticRegression | 0.30177 | 0.57200 | 0.76647 |

I choosed not to tune Gaussian Naive Bayes because this estimator does not have parameters to work with, just priors.

I got three models that stood out, both KNeighbors and NearestCentroid had a good performance when compared to others, but the chosen one was AdaBoost which had precision values similar to KNeighbors and NearestCentroid, but a higher recall value.

1. 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”]

With the gridsearcv function an exhaustive search of the best parameter for the estimator is done, sometimes in this search for the best model you set a value that increase the Recall metric and decrease Precison.

To be successful in this task the range of parameters must be carefully selected, in my test I had cases where my selection of parameters was incorrect and the tuning model returned metrics worse than its version with default parameters.

With only 18 POI records in the data set, there are few examples for learning. I tried reducing the dimensionality of the data with PCA and selecting the features with SelectkBest, but there was no improvement in the performance of the model as I would like. So I used MinMaxScale to transforms features by scaling then to a range between 0 and 1.

1. 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”]

Eu comecei este projeto usando a métrica F1 como referencia, mas ela é um balanço entre precision e recall e acabava retornando um valor alto quando uma das medidas era alta e eu precisava que tanto recall quanto precision fosse superior a 0.3.

Este projeto focou no precision (a fraction of positive labeled were correct), recall (a fraction of posivite labeled selectc correct) and accuracy (proportion of correct results), porque se eu tivesse investigado somente este valores poderia ter selecionado um classificador com baixa acurácia como foi o caso do Naive Bayes.

Outro ponto passível de falha é a escolha do tamanho da amostra que deve ser usada para treino e teste. Este dataset é pequeno e um dividir a amostra de treino em um pequeno percentual da amostra poderia prejudicar o modelo uma vez que não teria exemplos suficientes.

I split the dataset using cross\_validation.train\_test\_split and the test set size was 30%.

1. 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”]

Escolhi trabalhar com as métricas de recall e precision, mas sempre com a accuracy dando uma confiabilidade para o resultado.

Precision mede a frequência com que o modelo consegue identificar um POI no grupo de dados. O Recall me fala o percentual das vezes em que um registro apontado como POI era realmente um POI.

Identificar se um determinado funcionário está relacionado a uma esquema de fraude é uma tarefa bem importante e que se classificada incorretamento pode trazer sérias consequências para as pessoas listadas.

O modelo escolhido após o tuning foi o que retornou os valores mais equilibrados de precision, recall e accuracy.

Valores obtidos:

|  |  |  |  |
| --- | --- | --- | --- |
| Classifier | Precision | Recall | Accuracy |
| AdaBoostClassifier before tuning | 0.418319 | 0.3060 | 0.850733 |
| AdaBoostClassifier after tuning | 0.36377 | 0.63950 | 0.80280 |

Como você pode perceber após o tuning o modelo perdeu um pouco da accuracy e da precision, mas obteve um aumento significativo no valor de recall. O que é bastante significativo para mim porque me diz que o modelo melhorou bastante sua capacidade de apontar corretamente um POI.