Enron Submission Free-Response Questions

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 goal of this project is to build a good classifier that identifies people of interest in the Enron case. By good, I mean that it has to classify data with high accuracy and it also needs to have low training and testing time. Considering that it is a complex problem and it can contain many variables, machine learning is useful to accomplish it because it is a versatile technique since it can solve problems with linear decision surface and also nonlinear decision surface.

The data contains information about Enron employees, financial information, and e-mail information. Financial information includes salary, bonus, total stock value, etc, and all financial indicators are measured in dollars. E-mail information includes the total messages sent, total messages received, the messages that this person sent to a Person of Interest, and others. Person of Interest is my response variable and it is a boolean variable. The dataset contains a number of Person of Interest instances that are people involved in the Enron fraud.

I identified some outliers, but many were POI, so I could not remove them. The only data points I removed were TOTAL, THE TRAVEL AGENCY IN THE PARK, and LOCKHART EUGENE E. TOTAL seems to be the sum of all data points, THE TRAVEL AGENCY IN THE PARK does not seem to be an employee and LOCKHART EUGENE E contains only NaN. I plotted some scatterplots to identify outliers and some of them were a POI. For example, in the scatterplot of salary and bonus, I found two outliers, LAY KENNETH L and SKILLING JEFFREY K, but they are both POI. So, I thought it was not wise to exclude these outliers.

The dataset contains 146 data points, and after excluding TOTAL, THE TRAVEL AGENCY IN THE PARK and LOCKHART EUGENE E, there are 18 POI and 125 non-POI. Some features have a lot of missing values, as loan_advances, 140, director_fees, 127, restricted_stock_deferred, 126, and deferral_payments, 105. The variable total_stock_value has the small number of missing values, 18.

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"]

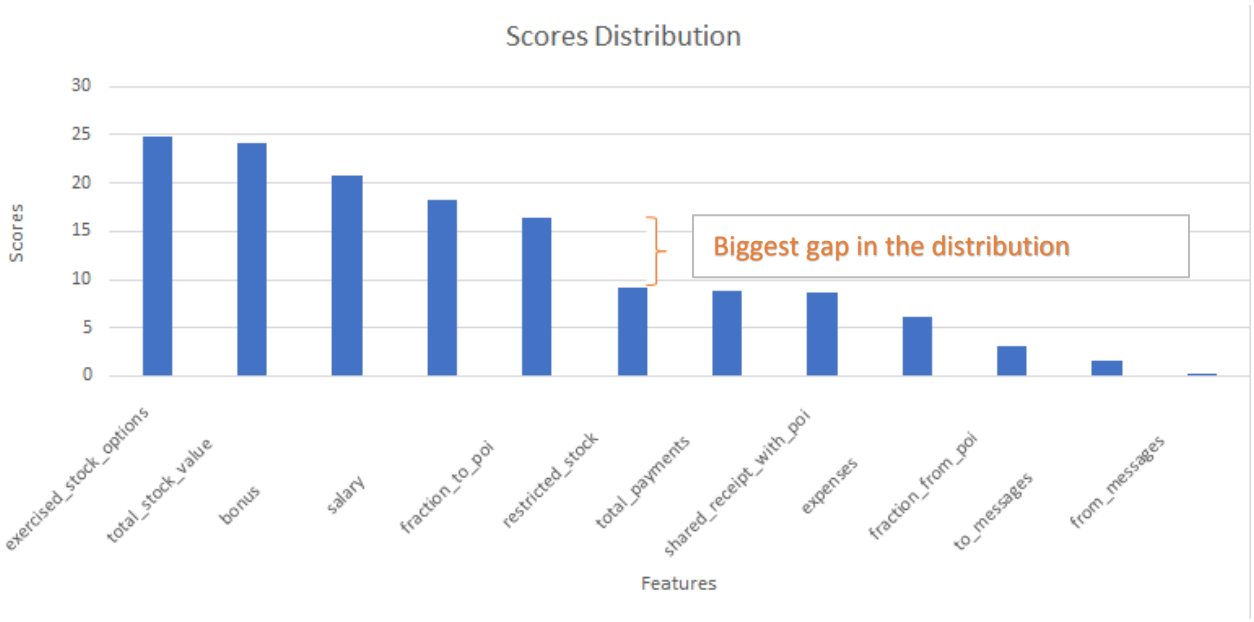
Before using a feature selection function, I plotted some scatterplots and analyzed some outliers. During this process, I thought that the number of messages that a person sends or receives it is not so important, but what matters it is if this person exchange emails a lot with a POI. So, I created two features: fraction_to_poi and fraction_from_poi. They identify the fraction of emails sent to and received from a POI.

In order to select my features, I used an automated feature selection function, SelectKBest. I started testing the features: salary, bonus, to_messages, shared_receipt_with_poi, fraction_to_poi, expenses, total_stock_value, total_payments, exercised_stock_options, restricted_stock, from_messages and fraction_from_poi.

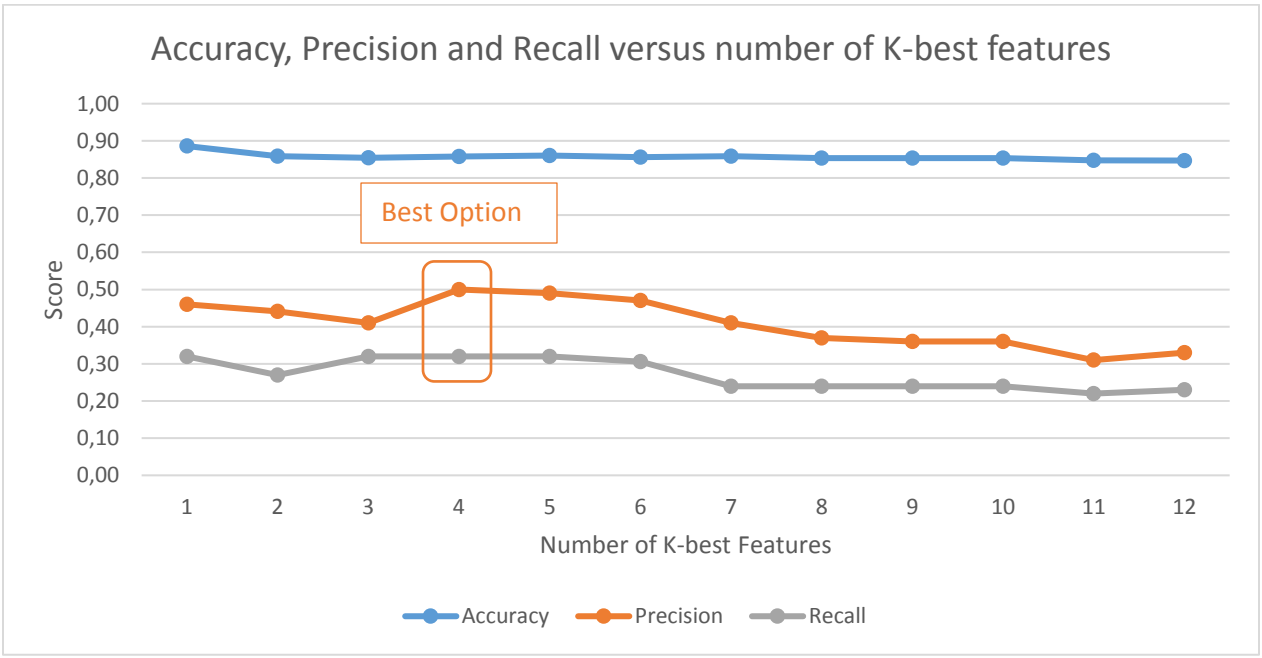
The scores of all variables evaluated are listed below and they are sorted descending.

[exercised_stock_options]	24.81507973
[total_stock_value]	24.18289868
[bonus]	20.79225205
[salary]	18.28968404
[fraction_to_poi]	16.40971255
[restricted_stock]	9.21281062
[total_payments]	8.77277773
[shared_receipt_with_poi]	8.58942073
[expenses]	6.09417331
[fraction_from_poi]	3.12809175
[to_messages]	1.64634113
[from_messages]	0.16970095

A distribution of these scores is plotted below.



To choose the best variables, I made a plot of the accuracy, precision and recall for many possibilities. The horizontal axis represents the number of features, so, the number one means that I used only the feature with the highest score, exercised_stock_options. This model gives a precision of 0.46 and a recall of 0.32. The number two represents a model with two variables, the two highest scores, exercised_stock_options, and total_stock_value. This model gives a precision of 0.44 and a recall of 0.27. The number 12 means that I used all the features.



Considering the graph, the model with the highest precision and recall contains only four variables. This way, my final feature list has the four variables with the highest score: exercised_stock_options, total_stock_value, bonus and salary.

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 tried six algorithms: Naïve Bayes, Decision Tree, SVM-Linear, SVM-RBF, KNN and Logistic Regression.

Naïve Bayes had training and testing time very fast, 0.001 and 0.0 seconds respectively, the average precision was about 0.59 and the average recall 0.36. Decision Tree had training and testing time very fast, 0.001 and 0.0 seconds respectively, the average precision was about 0.33 and the average recall 0.30. In the SVM-Linear case, I tried to run this classifier and after 10 minutes it didn't finish the training, so I interrupted the kernel and conclude that it is no a good classifier for my project. SVM-RBF had training and testing time also very fast, 0.001 and 0.0 seconds respectively. However, the model did not result in any true positive, so the precision and recall could not be calculated. The same happened with the KNN algorithm. The logistic regression had a precision of 0.13 and a recall of 0.03.

Considering the evaluation metrics above, I chose Naïve Bayes as the best algorithm for my project, because the training and testing time are fast and it had the highest average precision and recall.

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"]

Tuning the parameters of the algorithm is the process of trying different sets of parameters in order to get the best performance. If you do not tune your algorithm well, you could end up with a model that seems to perform well but is actually providing false results. For example, in the KNN algorithm, you need to set the number of neighbors. In my first attempt, I tried 10 neighbors, but I could have tried other values.

I chose Naïve Bayes as my classifier, so there were no parameters to tune.

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 helps us to estimate performance and also identify overfitting. If you do it wrong you can have a model that has a good precision, but it will fail to predict correctly new data. Considering that we have a class with the majority of the data, if we do not use cross validation, the algorithm will be biased, frequently predicting the majority class. In this dataset, there are many non-POI and just a few POI. So, if I do not use cross-validation, the algorithm will predict more frequently non-POI.

In order to validate my model, I used the Stratified Shuffle Split method. Also, 30% of the data is in the testing set, 70% is in the training set and the number of splits is 1000.

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 precision means how much of the data were classified correctly. In this model, about 59% of the instances were classified correctly.

The recall means the probability of the model classify an instance correctly. In this model, the probability of classifying an instance correctly is 0.36.