Enron Submission Free-Response Questions

A critical part of machine learning is making sense of your analysis process and communicating it to others. The questions below will help us understand your decision-making process and allow us to give feedback on your project. Please answer each question; your answers should be about 1-2 paragraphs per question. If you find yourself writing much more than that, take a step back and see if you can simplify your response!
When your evaluator looks at your responses, he or she will use a specific list of rubric items to assess your answers. Here is the link to that rubric: Link to the rubric Each question has one or more specific rubric items associated with it, so before you submit an answer, take a look at that part of the rubric. If your response does not meet expectations for all rubric points, you will be asked to revise and resubmit your project. Make sure that your responses are detailed enough that the evaluator will be able to understand the steps you took and your thought processes as you went through the data analysis.
Once you've submitted your responses, your coach will take a look and may ask a few more focused follow-up questions on one or more of your answers.
We can't wait to see what you've put together for this project!

1. **Summarise 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 the project is to build a prediction model to identify POI from non POI in Enron dataset given some features.Machine learning is very useful here as we can use its techniques to detect the underlying pattens for POI from the given features.

Most important characteristics:

There are 146 data points on the data set with 21 features for each person. 18 POIs in the dataset using the names file (poi_names.txt).For nearly every person in the dataset, not every feature has a value denoted with NaN except for POI which has either true or false in its value for all the data points.95 persons have a quantified salary and 111 persons have known email address. (about 14%) of the people in the dataset don't have total_payments filled in. 0% of POI's don't have total_payments filled.

outliers:
using salary and bonus features to detect outliers in the dataset reveals one outlier [Total] which must be removed because it is a spreadsheet quirk.after removing [Total] there were almost 4 more outliers including LAY KENNETH L and SKILLING JEFFREY K which seem to be valid data points so we should leave them in.

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"]**

I have used SelectKBest to select the best 5 features
to be used in my POI identifier and here is a list of the best 5 features with their scores.

Table 1: final best 5 features with scores	
salary	14.45
shared_receipt_with_poi	9.31
deferred_income	7.36
restricted_stock	10.11
total_payments	62.94

I didn't have to do feature scaling in my final work because I used Gaussian naive bayes algorithm which is not affected by feature scaling but I used it with SVM (without feature scaling, all the predictions were NON POI [0] resulted in an error on tester.py " Got a divide by zero").

I think that the financial data reveals the POI so I have added a new feature (total_payments_and_stock) represents person's total payments and stock value (total_payments + total_stock_value) which has an effect on both accuracy and validations as shown on table no. 2

Table 2: Effect of adding new feature			
	accuracy	precision	recall
before adding total_payments_and_stock feature	0.84	0.35	0.23
After adding total_payments_and_stock feature	0.85	0.43	0.32

I have chosen 5 an a parameter to SelectKBest because I have tested other numbers like 10, and 15. 5 almost provides the best result when used with Gaussian naive bayes algorithm as shown on the following section.

Previous work on this section:

I have started by passing 5 as a parameter for SelectKBest to get the best 5 features and the results was as below.

Table 3:best 5 features including loan_advances	
salary	14.45
shared_receipt_with_poi	9.31
loan_advances	8066.59
restricted_stock	10.11
total_payments	62.94

using these features with gaussian naive bayes algorithm results in the following performance

Table 4: performance of gaussian naive bayes with 5 features		
accuracy	precision	recall
0.74	0.10	0.11

The features in table 3 are same as table 1 except for the third feature (loan_advances).the score of it seems weird for me it is very high compared to the other features. So I decided to check it manually on the pdf file. I found that almost all persons have no value for this feature except 2 or 3 persons So I tried removing it from the total 21 features and rerun again SelectKBest to get the best 5 features excluding loan_advances and the result was the list shown in [table 1] and the performance improved very well [table 2]

In SelectKBest parameters I have tested many numbers like 5, 10 and 15 with gaussian naive bayes algorithm to test algorithm's performance on different groups of features.here is the result [table 5]

Table 5: performance of gaussian naive bayes with different number of features			
	accuracy	precision	recall
best 5 features	0.84	0.35	0.23
best 10 features	0.83	0.33	0.25
best 15 features	0.83	0.33	0.26

As shown in table 5 passing the best 5 features to gaussian naive bayes algorithm results in the best accuracy and precision.in recall value is 0.03 lower than the best one here (using 15 features)which I think we can ignore in opposite to the values of both accuracy and precision and definitely the cost of training with extra 10 features.

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"]

Three classification algorithms were tested to identify POI:

1- gaussian naive bayes
2- Decision tree
3- SVM

Table 6: algorithms' performance comparison

	accuracy	precision	recall
gaussian naive bayes	0.85	0.43	0.32
Decision tree	0.87	0.59	0.12
SVM	0.87	0.47	0.10

Table 6 shows the best performance for 3 different classification algorithms that I could reach.both decision tree and SVM have higher accuracy and precision but recall value is very low (under 0.3).So I have used gaussian naive bayes as its performance is accepted for the final project, both precision and recall exceed 0.3.

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"]

parameter tuning means trying different values for each parameter of an algorithm(if it has parameters) to get the best performance of the algorithm.If parameter tuning is not done well, the best performance of the algorithm will be missed.

In my try with decision trees I have used GridSearchCV for parameter tuning with both of min_samples_split and criterion parameters and the best estimators were 100 and gini respectively with SVM I have tried tuning gamma and C manually leaving kernel rbf. some of these tries are shown in table 7

Table 7: performance of SVM with parameter tuning

C	gamma	accuracy	precision	recall
10.0	0.1	0.86	0.10	0.0
100.0	0.1	0.86	0.46	0.11
1000.0	0.01	0.87	0.48	0.10
100.0	0.01	0.86	0.16	0.01

The best parameter tuning here for SVM was (C=1000.0, gamma=0.01).

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 assessing the performance of the algorithm on independent dataset by splitting dataset into training and testing datasets.if we don't split the dataset into two datasets (training and testing) and used all the available data in training the classifier then take part of it for evaluation we will get high score but will fail to predict new data (overfitting).also the absence of randomisation or shuffling dataset before splitting dataset into testing and training dataset can cause a wrong low accuracy.As you may train the algorithm on dataset with a specific pattern and test on another dataset with a different pattern which causes a bad accuracy.

I have validated my analysis using test_classifier function in tester.py which deployed cross validation within an iterative context to ensure that each data point got the chance in testing process so that validation results generalize well along dataset and don't depend on any specific pattern or characteristics of the dataset.it is also a good solution for overfitting problem which may occur.

In our problem we have unbalanced small dataset only 18 POIs and 130 non-POI. we want to preserve the ratio of POI and NON-POI So stratified shuffle split is used here by which samples are first shuffled and then split into train and test sets returning stratified splits preserving the same percentage for each target class [POI and NON-POI] as in the complete set.

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"]**

I have used 2 evaluation metrics precision and recall to get an overall view of the algorithm's performance.recall was 0.32 meaning that the probability of the algorithm to correctly classify a person as POI provided that the person actually is POI is 0.32. precision was 0.43 which is the probability of the person being an actual POI if it is classified as a POI