Answers to Machine Learning Project 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 Enron case is the largest story of corporate fraud that led to many people losing their job and the company going bankrupt. The data set contains data of individuals who were employed to Enron or connected to the company. Its contains financial data, email data from 163 individuals, and identifies whether a certain individual is a POI of interest. The dataset contains 21 features that we can use to help us train a machine learning algorithm to identify POIs which is the goal of this project. There were some outliers in the data set, namely a "TOTAL" entry which was a sum of every feature for all individuals. Once this was identified, the outlier was removed and I continued with the project.

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"]*

   To start off my feature selection process, I decided to separate the financial data from the email data and find the top features from each group. In order to find the top features from each group, I used SelectKBesta as my method to score each feature. I used the 'all' parameter for K to ensure I get a score for every feature. I picked the top 4 features from each group and combined them into another list. I executed SelectKBest again using k='all' to get new scores for the new feature list to explore the importance of each feature from the combined group. After going through this process, the features I used are below:

Feature	Score
exercised_stock_options	21.71552656
total_stock_value	21.05899501
bonus	17.8573624
salary	15.14904119

   Just to check for consistency, I decided to run SelectKBest on the entire list of features to confirm if these 4 features were the top 4 from the entire list. I also ran

SelectPercentile as well. After running SelectKBest and SelectPercentile on the entire feature list, I confirmed my initial selections were the top four. The entire feature scores can be seen below.

Feature	Score
exercised_stock_options	21.71552656
total_stock_value	21.05899501
bonus	17.8573624
salary	15.14904119
deferred_income	11.59554766
long_term_incentive	10.07245453
restricted_stock	9.34670079
total_payments	8.86672154
shared_receipt_with_poi	8.74648553
loan_advances	7.2427304
expenses	6.23420114
from_poi_to_this_person	5.34494152
other	4.20497086
from_this_person_to_poi	2.42650813
director_fees	2.10765594
to_messages	1.69882435
deferral_payments	0.21705893
from_messages	0.1641645
restricted_stock_deferred	0.06498431

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"]*

Since I didn't have a particular algorithm to try in mind, I chose to iterate through several classifiers to evaluate different metrics without making any parameter tunes to get a baseline of how each classifier performs. This will help me choose which classifier to focus tuning parameters on. When iterating, I captured the accuracy of a feature train/test split as well as a mean accuracy when running a 10-fold cross validation test to compare the two scores. I felt it was necessary to perform 10-fold cross validation to ensure all data is included in a test and a training procedure. Additionally, I dumped out the classifier, dataset, and feature list in each iteration so I can call tester.py. I recorded all the scores the tester file provides for each classifier. After iterating through each classier, I observed the following metrics:

Classifier	Tester Accuracy	Precision	Recall	Mean 10 Fold CV Accuracy	F1 Score	True positives	False positives	False negatives	True negatives
Decision Tree	0.791	0.326	0.336	0.8	0.331	672	1392	1328	9608
Naïve Bayes	0.847	0.50312	0.323	0.855	0.393	646	638	1354	10362
K nearest Neighbor	0.866	0.66	0.269	0.871	3.82	537	277	1463	1073
Random Forest	0.842	0.472	0.223	0.854	0.302	443	495	1557	10505
SVC	N/A	N/A	N/A	0.862	N/A	N/A	N/A	N/A	N/A

Due to the drawbacks accuracy can have when evaluating a classifier, I am choosing to focus on Precision and Recall. The classifiers that have the highest precision are:

- Naive Bayes
- K Nearest Neighbor

The classifiers that have the highest recall are:

- Decision Tree
- Naive Bayes

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”]*

   Parameters can make a huge difference in coming up with decision boundaries in classifiers. Tuning parameters means that you can set certain parameters in a given classifier to help control overfitting and help improve model performance. If this is not done correctly, your models can end up overfitting the data and can lead to inaccurate predictions and degraded performance. The algorithms and parameters I tuned can be found below

   - Decision Tree
     - 'criterion':('gini','entropy')
     - 'splitter':('best','random')
     - 'min_samples_split':[2,50,100,1000]
     - 'max_features':[1,2,3,4]
   - Random Forest
     - 'min_samples_split':[2,50,100]
     - 'n_estimators':[5,10,50,100]
   - K Nearest Neighbors
     - 'n_neighbors'[1-10]

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”]*

   Validation is a technique used in machine learning to make sure an algorithm is performing the way you want it to perform. We perform validation by splitting our data into training and testing data. This gives estimates of performance on an independent dataset and serves as a check on overfitting. A mistake you can do in validation is call a fit function on a test data set. We want to look for patterns in training sets and then use the test sets to validate steps we take when fitting data.

   I used a combination of Train/Test split and K fold cross validation to validate my analysis. I used a combination of both so I could compare how each validation performed against each other.

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"]*

   After exploring several different classifiers with parameter tunes, I decided to use K Nearest Neighbor as my classifier. The 2 evaluation metrics I used for this classifier are below:

   - Recall
     - Average recall = .4
     - Having good recall means that nearly every time a POI shows up in my test set, I am able to identify him or her. The cost of this is that some non-POIs get flagged.
   - Precision
     - Average recall = .433
     - Having good precision means that whenever a POI gets flagged, I know with confidence that it's very likely to be a POI. The cost of this is that I sometimes miss real POIs.