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”]

   In 2000, Enron was one of the largest companies in the United States. By 2002, it had collapsed into bankruptcy due to widespread corporate fraud. 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. The goal of this project is to employ machine learning skills in building a person of interest identifier based on financial and email data made public as a result of the Enron scandal. The aim is to classify Enron employees into two categories: POI and non-POI.

   The dataset contained 146 records with 1 labeled feature (POI), 14 financial features, 6 email feature. Within these record, 18 were labeled as a "Person Of Interest" (POI).

   Through exploratory data analysis and manual check, I find that there are three records to be removed.

   TOTAL: By using scatter-plot, I found TOTAL are the extreme outlier since it comprised every financial data in it.

   THE TRAVEL AGENCY IN THE PARK: This must be a data-entry error that it didn't represent an individual.

   LOCKHART EUGENE E: This has no non NaN values.

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”]

First, after remove the feature of email_address, I have the following feature_list.
features_list = ['poi','salary','to_messages', 'deferral_payments','total_payments','exercised_stock_options','restricted_stock', 'bonus', 'deferred_income', 'total_stock_value', 'expenses', 'exercised_stock_options','restricted_stock_deferred', 'loan_advances','other','director_fees','shared_receipt_with_poi','from_messages','long_term_incentive','from_poi_to_this_person']

Second, I create two new features,” from_poi” and “to_poi”.

from_poi= from_this_person_to_poi/ to_messages

to_poi= from_poi_to_this_person// from_messages

Third, I scaled all features using the scikit-learn MinMaxScaler to avoid problems caused by different units in the dataset.

Finally, I applied SelectKBest in sklearn to select the K=5 best features from features_list. The feature scores and selected features are as follows.

salary	0.000160054245696l8399	False
to_messages	1.7516942790340737	True
deferral_payments	0.23899588985313305	False
total_payments	0.34962715304280179	False
exercised_stock_options	0.22826733729104948	False
restricted_stock	0.031333216297618476	False
bonus	0.077948855777229875	False
deferred_income	0.21950572394230994	False
total_stock_value	0.16611912320976677	False
expenses	0.01397841382175243	False
exercised_stock_options	0.22826733729104948	False
restricted_stock_deferred	0.0041731922805086684	False
loan_advances	2.5182610445203437	True
other	0.068194519159558625	False
director_fees	0.54908420147980874	True
shared_receipt_with_poi	8.9038215571655712	True
from_messages	0.1587702392129193	False
long_term_incentive	0.022229270861607336	False
from_poi_to_this_person	5.4466874833253529	True

Therefore, the selected features are ‘to_messages’, ‘loan_advances’, ‘director_fees’, ‘shared_receipt_with_poi’ and ‘from_poi_to_this_person’.

In the Table below, I list the performance scores for the case of all the features and the selected features.

	all the features	the selected features
accuary score of naive_bayes	0.741379310345	0.788461538462
precision score of naive_bayes	0.413461538462	0.418367346939
recall score of naive_bayes	0.438775510204	0.465909090909
accuary score of DecisionTreeClassifier	0.741379310345	0.788461538462

precision score of DecisionTreeClassifier	0.685185185185	0.416666666667
recall score of DecisionTreeClassifier	0.590702947846	0.454545454545

It is shown that the accuary scores of using the selected features are a little higher than the case of using all the features for both naive_bayes and DecisionTreeClassifier. However, this is not the case for precision score and recall score. The precision score and recall score of using the selected features are lower than case of using all the features for both naive_bayes and DecisionTreeClassifier. Due to the nature of our dataset, i.e., unbalanced and small, in this project, precision and recall score are more reliable performance measure. Therefore, in trying algorithms, I end up using all the 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"]

   I have tried two algorithms which are naive_bayes and DecisionTreeClassifier. Their performance in terms of accuary, precision and recall are listed as follows.

	naive_bayes	DecisionTreeClassifier
accuray score	0.741379310345	0.741379310345
precision score	0.413461538462	0.685185185185
recall score	0.438775510204	0.590702947846

It is shown that DecisionTreeClassifier has a better precision and recall performance. Therefore, I choose to use DecisionTreeClassifier.

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 an algorithm can be simply thought of as process which one goes through in which they optimize the parameters that impact the model in order to enable the algorithm to perform the best. Tuning a machine learning algorithm is crucial because different functions and initial settings can have a profound effect on its performance. In some cases, such as selecting a wrong minimum number of samples per leaf in a Decision Tree algorithm, the algorithm can overfit. In other cases, such as selecting the wrong number of clusters for a KMeans algorithm, the end result can be entirely wrong and unuseable.

   I performed automatic parameter tuning using scikit-learn GridSearchCV during the algorithm selection process. The tuned parameters are shown below.

```
param_grid = {"selection__k": range(4,10),
 "dt__criterion": ["gini", "entropy"],

"dt__min_samples_split": [2, 10, 20],

"dt__max_depth": [None, 2, 5, 10],

"dt__min_samples_leaf": [1, 5, 10],

"dt__max_leaf_nodes": [None, 5, 10, 20],

}
```

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”]

   Before applying the built algorithms in practice, we need to have some way of accessing whether the algorithm is actually doing what we want it to do and how well it is doing.

   Validation is to estimate how well the model has been trained and to estimate model properties. Therefore, we use validation in the first place after building the model.

   The benefits of using training & testing data are as follows. First, give estimate of performance on an independent dataset. Second, serve as check on overfitting.

   The validation phase is often split into two parts:

   In the first part you just look at your models and select the best performing approach using the validation data (=validation)

   Then you estimate the accuracy of the selected approach (=test).

   Due to the imbalanced and small data we have in this project, I choose to use StratifiedShuffleSplit instead of KFold for the cross validation.

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”]

   Below is the final result of my algorithm. The explanations of recall and precision are as follows.

   ```
   Accuracy: 0.81313       Precision: 0.32195      Recall: 0.36300 F1: 0.34125     F2:
   0.35397

           Total predictions: 15000        True positives:  726    False positives: 1529   False
   negatives: 1274   True negatives: 11471
   ```

Recall is the measure of the probability that your estimate is 1 given all the samples whose true class label is 1. It is a measure of how many of the positive samples have been identified as being positive.

Recall= True positives/(True positives+ False negatives)

Plugging the values of True positives and False negatives into the equation of Recall, we have

Recall=726/(726+1274)=0.363

Precision on the other hand is different. It is a measure of the probability that a sample is a true positive class given that your classifier said it is positive. It is a measure of how many of the samples predicted by the classifier as positive is indeed positive.

Precision = True positives/(True positives+ False positives)

Similarly, plugging the values of True positives and False positives into the equation of Precision, we have

Recall=726/(726+1529)= 0.32195