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"]

Enon Dataset Overview:

Enron Corporation was an American energy, commodities, and services company based in Houston, Texas. It was founded in 1985 as the result of a merger between Houston Natural Gas and InterNorth.[1] Fortune named Enron "America's Most Innovative Company" for six consecutive years. At the end of 2001, it was revealed that its reported financial condition was sustained by institutionalized, systematic, and creatively planned accounting fraud, known since as the Enron scandal. Enron has since become a well-known example of willful corporate fraud and corruption. [2]

In this project I will use machine learning algorithms to create a predictive model to help identify Enron's persons of interest, who were involved in the scandal.

The dataset consist of two main categorical features: Financial features, and Email features, in addition to the hand-curated labels of whether this record belong to POI or not:

- **Financial features:** 'salary', 'deferral_payments', 'total_payments', 'loan_advances', 'bonus', 'restricted_stock_deferred', 'deferred_income', 'total_stock_value', 'expenses', 'exercised_stock_options', 'other', 'long_term_incentive', 'restricted_stock', 'director_fees'
- **Email features:** 'to_messages', 'email_address', 'from_poi_to_this_person', 'from_messages', 'from_this_person_to_poi', 'shared_receipt_with_poi'
- **POI labels:** 'poi'

The dataset is very limited it contains [146] data points and only [18] are identified POI. This will be challenging and we will need cross-validation and careful selection of features and validation matrix to verify the validity of our predictive model.

Dealing with Outliers:

Creating simple visualization of the data can provide us with view of outliers, will then investigate these records and use human intuition to conclude if this is acceptable or not.

There is indeed one record record 'TOTAL' that is definitely an outlier so it will be removed, the rest of the data points look normal enough to be kept in our dataset. However, there is also one record that has no values for any of the features so it will be automatically removed with 'FeatureFormat' method provided by passing (True) for the parameter 'remove_all_zeros', which will basically do is that it will omit any data points for which all the features you seek are 0.0, this record belongs to 'LOCKHART EUGENE E'. There is also a record that does not belong to an actual person 'THE TRAVEL AGENCY IN THE PARK', so it will be removed as well.

Final Dataset Overview:

- **Total dataset count:** 143
- **POI count:** 18
- **Non-POI count:** 125

- **Features with missing values:** unfortunately the dataset contains many missing values:
  [('loan_advances', 141), ('director_fees', 128), ('restricted_stock_deferred', 127), ('deferral_payments', 106), ('deferred_income', 96), ('long_term_incentive', 79), ('bonus', 63), ('to_messages', 58), ('from_poi_to_this_person', 58), ('from_messages', 58), ('from_this_person_to_poi', 58), ('shared_receipt_with_poi', 58), ('other', 53), ('salary', 50), ('expenses', 50), ('exercised_stock_options', 43), ('restricted_stock', 35), ('email_address', 33), ('total_payments', 21), ('total_stock_value', 19)]

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 will start my model building with all features provided. Using Pipeline to perform these steps of feature engineering: feature scaling, automated feature selection, and finally provide dimensionality reduction with PCA. Feature scaling to limit [0,1] is required because there are different units for features, for example: hundred of thousands USD, and count of emails. For the automated selection will use (SelectKBest) with various parameters and select the ones that give best performance based on score matrix (f_classif). (PCA) dimensions reduction have given better results so we will keep it in our model.

New Feature Engineering:
I have added two new features, a fraction of emails from POI, and fraction of messages to POI. I assume that they can be valuable in identifying POIs, since the Enron's emails data were of high value during the investigation, these feature were thought of because having ratio of message sent to\from POI could also indicate the other person is POI. Adding these feature and testing the model initially gave better results, so I decided to keep them and have the selection algorithm rule them out if needed. Below is sample comparison between the model with provided features and with new added features, using one of the algorithms I intend to test later on:

Algorithm	Features Selected (SelectKBest)	Accuracy	Precision	Recall	F1
LinearSVC	**Original features:** ['salary', 'deferral_payments', 'loan_advances', 'bonus', 'deferred_income', 'total_stock_value', 'expenses', 'exercised_stock_options', 'other', 'long_term_incentive', 'restricted_stock', 'director_fees', 'to_messages', 'from_poi_to_this_person', 'from_messages', 'from_this_person_to_poi', 'shared_receipt_with_poi']	0.73333	0.19916	0.33100	0.24869
	With new added features: ['salary', 'deferral_payments', 'bonus', 'deferred_income', 'total_stock_value', 'expenses', 'exercised_stock_options', 'other', 'long_term_incentive', 'restricted_stock', 'director_fees', 'fraction_from_poi', 'fraction_to_poi']	0.73727	0.20475	0.33650	0.25459

SelectKBest Results:
I have tried different set of values for different algorithm, an overview of how well the features performed will be provided with the algorithms comparison, and the values tested during tuning will be discussed later on.

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 this is straightforward classification problem, I will try out these algorithms:

- Support Vector Machine (SVC, and LinearSVC).
- Decision Tree Classifier.
- Random Forest Classifier.

In the next section I will explain in details how these algorithms were tuned in order to find the optimal algorithm and its best parameters.

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 item: "tune the algorithm"]

Algorithms Tuning:
Parameters of each algorithm have an important factor on how it will perform. Tuning an algorithm is to try different values based on the structure of the data in order to find optimal results. This task is very critical because we may end up over-tuning the algorithm this would result in the model predicting very well on train data but poorly on the test data.

To achieve optimal results the parameters available for each algorithm will be tuned using GridSearchCV and since the dataset has small number of datapoint the search will on stratified cross-validated dataset. The process will be using Pipeline with these steps [scaling, selection, reduction, classification]. The results as follows:

Algorithm	**Select KBest Tuning**	**Optimal SelectKBest**	**Selected Features**	**Top Features Importance**	**PCA Tuning**	**Optimal PCA (n)**
SVC	**K**: [5, 7, 9, 13, 15, 17, 20, 'all']	**K**: 5	['salary', 'deferred_income', 'expenses', 'fraction_from_poi', 'fraction_to_poi']	Not applicable	**N**: [1, 2, 3, 4, 5, .25, .5, .75, .9, 'mle']	**N**: 1
LinearSVC	**K**: [5, 7, 9, 13, 15, 17, 20, 'all']	**K**: 21 'all'	['salary', 'deferral_payments', 'total_payments', 'loan_advan	Not applicable	**N**: [1, 2, 3, 4, 5, .25, .5, .75, .9, 'mle']	**N**: 2

			ces', 'bonus', 'restricted_stock_deferred', 'deferred_income', 'total_stock_value', 'expenses', 'exercised_stock_options', 'other', 'long_term_incentive', 'restricted_stock', 'director_fees', 'to_messages', 'from_poi_to_this_person', 'from_messages', 'from_this_person_to_poi', 'shared_receipt_with_poi', 'fraction_from_poi', 'fraction_to_poi']			
DecisionTreeClassifer	**K**: [5, 7, 9, 13, 15, 17, 20, 'all']	**K**: 15	['salary', 'deferral_payments', 'bonus', 'deferred_income', 'total_stock_value', 'expenses', 'exercised_stock_options', 'other', 'long_term_incentive', 'restricted_stock', 'director_fees', 'from_poi_to_this_person', 'from_messages', 'fraction_from_poi', 'fraction_to_poi']	Bonus: 0.238502478776	**N**: [1, 2, 3, 4, 5, .25, .5, .75, .9, 'mle']	**N**: 4

RandomForestClassifier	**K**: [5, 7, 9, 13, 15, 17, 20, 'all']	**K**: 9	['salary', 'deferred_income', 'total_stock_value', 'expenses', 'exercised_stock_options', 'long_term_incentive', 'director_fees', 'fraction_from_poi', 'fraction_to_poi']	fraction_to_poi : 0.306197443339	**N**: [1, 2, 3, 4, 5, .25, .5, .75, .9, 'mle']	**N**: 1

Algorithm	**Parameters Tuning**	**Optimal Parameters**	**Evaluation**			
			Accuracy	**Precision**	**Recall**	**F1**
SVC	**C**:[1,10.,100,1e3, 1e4, 1e5] **Kernel**: rbf **random_state**: [13, 20, 42]	**C**: 100 **Kernel**: rbf **random_state**: 13	Cross-validated recall score: 0.66 Got a divide by zero when trying out: SVC(C=100, cache_size=200, class_weight='balanced', coef0=0.0, decision_function_shape=None, degree=3, gamma=0.0, kernel='rbf', max_iter=-1, probability=False, random_state=13, shrinking=True, tol=0.001, verbose=False)			
LinearSVC	**C**:[1,10.,100,1e3, 1e4, 1e5] **random_state**: [13, 20, 42]	**C**: 1 **random_state**: 13	0.73847	0.21238	0.3550	0.2657
DecisionTreeClassifer	**min_samples_split**: [10, 15, 30, 40] **criterion** : ['gini', 'entropy'] **splitter**: ['best', 'random'] **random_state**: [13, 20, 42]	**Min_samples_split**: 15 **criterion** : 'gini' **splitter**: 'random' **random_state**: 13	0.74573	0.26874	0.5270	0.3559
RandomForestCalssifier	**min_samples_split**: [10, 15, 30, 40] **criterion** : ['gini', 'entropy'] **splitter**: ['best', 'random'] **random_state**: [13, 20, 42] **n_estimators**: [5, 7, 10, 20]	**min_samples_split**: 30 **criterion** : 'gini' **splitter**: ['best', 'random'] **random_state**: 42 **n_estimators**: [5, 7, 10, 20]	0.80180	0.36490	0.60500	0.45523

Final Algorithm Selection and Optimal Parameters:
The final model of choice will be using (Random Forest Classifier) since it gave the best results based on the evaluation matrixes: Precision and Recall.

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 the process of assessing the quality of a model on test data. A naive mistake would be overfitting which means that the model performs well on train data but poorly on test data. This issue is the result of when the whole dataset was used during the training process, or in some cases if we had a rather bad choice of splitting dataset into training and testing subsets.

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"]

Model Validation:
An important factor of the validation process is the Train/Test dataset split, for this dataset a classic holdout method will not be sufficient nor will it give much confidence in evaluation matrix, due to the small number of datapoints. Another alternative is to use Stratified Cross-Validation process which will run through predefined number of folds (1000) in each one it will randomly split the data into train\test subset, where the model will be trained and tested on, and an result of each split is calculated and the final result is the average of all iterations. This will overcome the problem we may encounter when using holdout method. Below is a comparison between the result of both methods, although the holdout method is performing better than cross-validation method, we can not be at all confident in its results due to the characteristics and structure of the dataset:

Split method	Accuracy	Precision	Recall	F1
test_train_split	0.837209302326	0.714285714286	0.5	0.5882352941
StratifiedShuffleSplit	0.80180	0.36490	0.60500	0.45523

Evaluation:
For the evaluation of this model I focused on two matrixes (Precision and Recall) rather than simple accuracy value. This is critical because since we have a limited number of datapoint and the ratio of non-POI to POI is high, meaning this would drive the model to predict non-POI and give good accuracy results. Whereas the Precision and Recall will pay attention to the actual flagging and identification of POI and non-POI and how it would compare to the rest of predictions.

Precision: can be simply put as the rate that, a person is actually POI if it was predicted to be POI.
Recall: corresponds to the rate that if a person is a POI then the model would predict it correctly.

All evaluation results are provided in the previous section with the algorithms comparison.