Intro to Machine Learning
Identity Fraud from Enron Email

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 is project is to find the person of interest in Enron using optimised machine learning algorithm. Using machine learning can help in identifying that if the person in question is of interest or not.
The data had both the financial features and email features of previous Enron employees.
Financial features include ['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'] (all units are in US dollars)
Email features include ['to_messages', 'email_address', 'from_poi_to_this_person', 'from_messages', 'from_this_person_to_poi', 'shared_receipt_with_poi'] (units are generally number of emails messages; notable exception is 'email_address', which is a text string)
POI label: ['poi'] (boolean, represented as integer)

While exploring data, I found three outliers, which I removed from the data.
First was the name in dataset 'LOCKHART EUGENE E', it had NaN as value for all the features.
And the other two were the data entry for 'TOTAL' and 'THE TRAVEL AGENCY IN THE PARK'.
TOTAL was the aggregate for all the financial features and 'THE TRAVEL AGENCY IN THE PARK' was not a valid Employee of Enron.
After removal of outliers the value for NaN was changed to 0.

In dataset I found out the number of missing values for the feature,

```
{'deferral_payments': 105, 'loan_advances': 140,
'restricted_stock_deferred': 126, 'deferred_income': 95,
'exercised_stock_options': 42, 'long_term_incentive': 78,
'director_fees': 127, 'to_messages': 57, 'from_poi_to_this_person': 57,
'from_messages': 57, 'from_this_person_to_poi': 57,
'shared_receipt_with_poi': 57, 'salary': 49, 'total_payments': 20,
'bonus': 62, 'expenses': 49, 'other': 52, 'restricted_stock': 34,
'total_stock_value': 18}
```

Total number of data points are 143 (After removing outliers)
18 people are marked as POI and **125** are marked as Non-POI

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"]**

I used three features exercised_stock_options, total_stock_value, bonus for the POI identifier. I used selectKBest to get the scores of the features and used value of k from 1 to all.
For k=3, algorithm had the best precision and recall score and highest f1 score.
For k=1, I got Precision: 0.42869 Recall: 0.26450 F1: 0.32715
k=2, Precision: 0.36048 Recall: 0.37850 F1: 0.36927
K =3, Precision: 0.36507 Recall: 0.41600 F1: 0.38888
k= 4, Precision: 0.30220 Recall: 0.28800 F1: 0.29493
As I increased the value okay k, the precision, recall and f1 scores started to decrease.

I engineered two new feature, from_poi_to_persion_ratio and from_this_person_to_poi_ratio
'from_poi_to_persion_ratio' = 'from_poi_to_this_person'/'from_messages', it gives the ratio between the messages sent to this person by POI by total messages received.
'from_this_person_to_poi_ratio' = 'from_this_person_to_poi'/ 'to_messages', it gives the ratio between the messages sent by the person to POI by total messages sent.
These features were not included into the final features, because their scores were very less (<0.007)

The following scores were obtained using SelectKbest,

	Feature Name	Scores
2	bonus	25.796753
1	exercised_stock_options	17.077582
0	total_stock_value	16.689335
7	shared_receipt_with_poi	13.042650
6	long_term_incentive	12.474234
8	restricted_stock	9.117468
3	from_poi_to_this_person	4.705397
9	director_fees	1.211803
4	from_messages	0.331777
5	from_this_person_to_poi	0.007671

With the help of above scores, top three features were considered for final algorithm

For scaling the features I used MinMaxScaler, PCA.
As KNeighborsClassifier relies on the distance, If one feature has wider range of values, it would be weighted more than any other feature while making decision.
Using scaling helps Im making other features weighted fairly.
And for DecisionTreeClassifier, as it does not rely on distance, scaling was not required.

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 compared two machine learning algorithms, KNeighborsClassifier and DecisionTreeClassifier. Below are the result of comparison,

DecisionTreeClassifier

```
Accuracy: 0.83069    Precision: 0.43935   Recall: 0.36400 F1: 0.39814
F2: 0.37693
    Total predictions: 13000 True positives:  728 False positives:  929
False negatives: 1272 True negatives: 10071
```

KNeighborsClassifier

```
Accuracy: 0.86308    Precision: 0.68272   Recall: 0.20550 F1: 0.31591
F2: 0.23890
    Total predictions: 13000 True positives:  411 False positives:  191
False negatives: 1589 True negatives: 10809
```

Based on above the above F1 scores, I used 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? 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"]**

Tuning parameters of an algorithms means using various values for the parameters passed in algorithms to get a combination that gives us optimal result.
Every machine learning algorithms have some keys called parameters which run them, Each parameter have a default value. By using default value for different types of datasets we can not get optimised result. We try different value for these parameters until we get the best optimised result.

For example the Decision tree classifier that is used in out algorithms, I used various values of parameters,

```
parameters = {'min_samples_split': [2, 3, 4, 5, 6, 7], 'max_features': ['auto', 'sqrt',
'log2', None], 'criterion': ['gini', 'entropy']}
```

In this I used different values of criterion, min_samples_split and max_features.
And below is the best combination that gave us best result.

```
DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=None,
                       max_features='log2', max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=6,
                       min_weight_fraction_leaf=0.0, presort=False,
                       random_state=None, splitter='best')
```

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 method of testing whether our algorithms is predicting as it should be. We split the data into two sets of training and testing sets. With training set we fit our algorithm and by testing we validate that the prediction made by our algorithm is correct or not.

The main mistake that is made is overfitting the data.

I used StratifiedShuffleSplit validation by splitting 70% of data in to training and 30% into test sets.
This cross-validation object is a merge of StratifiedKFold and ShuffleSplit, which returns stratified randomized folds. The folds are made by preserving the percentage of samples for each class.
The data has 125 Non POI and 18 POI that around 87% are Non POI, By using this validation method we shuffled the data so that training data should have both POI and Non POI for better fitting of data.
By using the training set I fit the algorithm and got predictions, and then compared the predictions to true values from labels_test set.

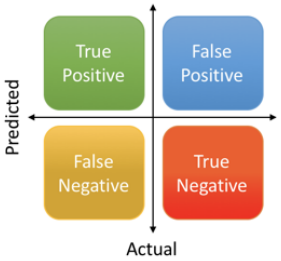
6. Give at least 2 evaluation metrics and your average performance for each of them. Explain an interpretation of your metrics that says something human-understandable about your algorithm's performance. [relevant rubric item: "usage of evaluation metrics"]

The evaluation metrics used were recall , precision and f1.
recall refers to the percentage of total relevant results correctly classified by the algorithm and precision means the percentage of the results which are relevant

$$\textbf{Precision} = \frac{\text{True Positive}}{\text{Actual Results}} \quad \text{or} \quad \frac{\text{True Positive}}{\text{True Positive + False Positive}}$$

$$\textbf{Recall} = \frac{\text{True Positive}}{\text{Predicted Results}} \quad \text{or} \quad \frac{\text{True Positive}}{\text{True Positive + False Negative}}$$

$$\textbf{Accuracy} = \frac{\text{True Positive + True Negative}}{\text{Total}}$$



F1 score is a combination of recall and precision.

$$\textbf{F1 Score} = 2 * \frac{\text{Precision * Recall}}{\text{Precision + Recall}}$$

The recall score was 0.36 and precision score was 0.43
F1 score was 0.39 (the higher the value of f1 score the better the algorithm)