Lucas Mwai
Udacity Data Analyst Nanodegree
Identify Fraud from Enron Email project

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 Corporation which was an American energy company based in Houston, Texas went bankrupt in 2011 which was one of the five largest audit and accountancy partnerships in the world. It was one of the biggest audit failures in American history at that time. A database of emails from employees of the Enron Corporation was acquired by the Federal Energy Regulatory Commission in order to investigate the company. Many executives at Enron were indicted for a variety of charges and some were later sentenced to prison
The goal of the project is to utilize machine learning to build a model that predicts where a person is a POI. The Enron email corpus is used in this project to identify people of interest in the Enron scandal. A "Person of Interest" is defined as:

- someone who was indicted.
- someone who settled without admitting guilt.
- someone who testified in exchange for immunity.

The dataset in this project is a dictionary combined from the Enron email and financial data, where each key-value pair in the dictionary corresponds to one person the features in the data fall into three major types, namely financial features, email features and POI labels. the POI label is the target feature that we want to predict, while the financial and email features are the input features. This project aims to find the relationship between the target feature and the input features.

```
There is a total of 146 people in the dataset.
There are 18 POIs and 128 Non-POIs.
Total number of email plus financial features are 20. 'poi' column is
the label.
```

The field "TOTAL" in this dataset as displayed below is not a record for an individual person but an aggregate record, it is a clear outlier from the scatter plot so it will be removed from the dataset
The name "THE TRAVEL AGENCY IN THE PARK", which does not correspond to an individual. This observation was therefore removed from the data set, another record of LOCKHART EUGENE contained only NaN data, so it will be removed from the dataset. That brings the number of features to be used for the project to 143

2.What features did you end up using in your POI identifier, and what selection process d id you us 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 y our feature selection step, if you used an algorithm like a decision tree, please also give t he feature importance's of the features that you use, and if you used an automated featur e selection function like SelectKBest, please report the feature scores and reasons for yo

ur choice of parameter values. [relevant rubric items: “create new features”, “intelligentl y select features”, “properly scale features”]

I used the following features

```
Optimized features list :
['poi',
 'exercised_stock_options',
 'total_stock_value',
 'bonus',
 'salary',
 'deferred_income',
 'long_term_incentive']
```

I used VarianceThreshold to remove the features that had a variance that fell below 80%, I also use d SelectKBest function to compute the scores then used the top 6 features that gave the best scores for the prediction model, I also tested for a variety of numbers of features but I settled for 6 which g ave the best combination of the relevant scores, I also used min-max scaler to linearly rescale the fe atures

The various K numbers are recorded as follows

k value	Accuracy	Precision	Recall	F1 score
k=5	0.874	0.601	0.351	0.441
k=6	***0.872***	***0.579***	***0.390***	***0.465***
k=7	0.861	0.514	0.373	0.432
k=8	0.871	0.524	0.390	0.447
k=10	0.863	0.479	0.322	0.385
k=12	0.862	0.472	0.280	0.357
k15	0.835	0.350	0.364	0.356

Although k=5 recorded a higher accuracy the precision and recall (F1 score) are better measures fo r this dataset because accuracy is not a good score in this case since the dataset in this project is ver y small and the ratio of negatives to positives is highly skewed, so k=6 has the best scores consideri ng the precision and recall

select**KBest features scores:**

```
[('exercised_stock_options', 24.815079733218194),
 ('total_stock_value', 24.182898678566879),
 ('bonus', 20.792252047181535),
 ('salary', 18.289684043404513),
 ('deferred_income', 11.458476579280369),
 ('long_term_incentive', 9.9221860131898225),
 ('restricted_stock', 9.2128106219771002),
 ('total_payments', 8.7727777300916756),
 ('shared_receipt_with_poi', 8.589420731682381),
 ('loan_advances', 7.1840556582887247),
 ('expenses', 6.0941733106389453),
 ('from_poi_to_this_person', 5.2434497133749582),
 ('other', 4.1874775069953749),
 ('from_this_person_to_poi', 2.3826121082276739),
 ('director_fees', 2.1263278020077054),
 ('to_messages', 1.6463411294420076),
```

```
('deferral_payments', 0.22461127473600989),
('from_messages', 0.16970094762175533),
('restricted_stock_deferred', 0.065499652909942141)]
```

I also used pipeline to put together the feature scaling and model training process. I made 2 new features named 'msg_from_poi_ratio' (the ratio a person receives emails from POI) and msg_to_poi_ratio (the ratio a person sends emails to POI). Since there was ongoing corruption it would be very likely that the POIs were in constant contact with other POIs to facilitate the communication that was facilitating the corruption, however, the scores I got from SelectKBest function did not validate this claim, the performance of the classifier was poorer as compared to the performance when these new features were included in the analysis, so I ended up not using them and used the features that scored much better as shown below . various classifiers performed better with various combination of features but ultimately Naïve Bayes did the best

```
Comparison of performance with/without new features with naïve Bayes c
lassifier
New features not included
Accuracy: 0.87243 Precision: 0.57938 Recall: 0.39050 F1: 0.46655
New features included
Accuracy: 0.86250 Precision: 0.52817 Recall: 0.35150 F1: 0.42210
```

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 tried out 4 algorithms Naïve Bayes, Logistic Regression SVM and Decision Trees. Naïve Bayes recorded the best performance based on the F1 score which is a balanced metric to measure both the precision and the recall scores, the summary of the performance is as follows

```
Naïve Bayes  Accuracy:0.8694 Precision:0.5620 Recall: 0.3895 F1: 0.4601
L Regression Accuracy:0.7352 Precision:0.2536 Recall: 0.4395 F1: 0.3216
SVM          Accuracy:0.6694 Precision:0.1995 Recall: 0.4365 F1: 0.2739
Dec Trees    Accuracy:0.7374 Precision:0.2100 Recall: 0.3035 F1: 0.2482
```

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 algorithm is a process of configuring parameters aiming to achieve the optimal combination of which the algorithm performs the best, the parameters configured are meant to enable the algorithm to perform the best and give the best possible results. Lack of tuning the algorithm results to the opposite result because without tuning there is no discovering the combination of parameters that result in the best performance

I used the GridSearchCV() to turn the parameters in order to try to get the best performance of each model. For naïve Bayes classifier there's no need to specify any parameters
An algorithm that would need some tuning is SVM, there are parameters that need to be tuned for optimal performance including _C: with a default of 1.0)

gamma': an optional float with default auto
degree an optional integer with default 3
kernel': default rbf with options linear and poly

tuning these parameters would involve trying out different parameters and scoring them for the best performance
For this project, the SVM parameters I tuned were `'svc__C': 5000, 'svc__gamma': 0.005` settling with defaults for other parameters

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 the strategy that is used on an algorithm to evaluate the performance of the model on unseen data. If training data is used to test the performance of a model, A classic mistake is to evaluate the performance of an algorithm on the same dataset it was trained on, it would give a perfect or close to a perfect performance. There is thus a need to test how good a model is using a different set of data different from the training set

I used stratifiedKFold to perform the cross validation in the parameters tuning process and used the test_classifier function provided in the tester.py script to test the model performance.

This being such a small dataset, the tester.py file uses StratifiedShuffleSplit instead of a simpler cross-validation method such as TrainTestSplit. StratifiedShuffleSplit will make randomly chosen training and test sets multiple times and average the results over all the tests it also makes sure that the ratio of non-POI:POI is the same in the training and test sets as it was in the larger data 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"]

Accuracy demonstrates the proportion a measured value compares to the actual value. An accuracy of 0.8694 as above indicates the model had about 87% match to both true positives and true negatives.

Precision is a metric that measures an algorithm's power to classify true positives from all cases that are classified as positives. A precision of 0.562 as above means that about 56% persons are actually POIs as identified in the classification,

Recall Score is the percentage ratio of the samples which were correctly predicted as the positive class of all the actual positive class as pertains to this project, the Recall Score is a measure of the ratio of the POIs that the algorithm correctly picked out from all the POIs in the dataset, in this case its 39%.

F1 score generally a model with high recall and precision scores is ideal, however because it's not possible to achieve high scores for both, thus the F1 score is used to aggregate the two scores.