Project P5 : Identify 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”]

Ans : The goal of this project is to build a system to identify/predict persons of interest. The dataset for this assignment is based on the financial and email metadata of the employees of Enron Corporation, an American energy company based in Houston, Texas. The company got bankrupt in October 2001. Besides being the largest bankruptcy organization in the American history at that time, Enron Corporation was cited as the biggest audit failure.

In this project, I intend to use supervised machine learning algorithm to predict whether a person is a person of interest involved in Enron fraud. To achieve that, a combination of features from the financial and email data of the employees of the company is used as an input. By training and testing the algorithm on separate training and testing sets, I will give examples for the algorithm to learn and capture the trends in the training data and then use its learning power to perform predictions on the testing data, to come up with discrete outcomes (POI/1 or Non-POI/0). The objective is to get the maximum accuracy possible with a precision & recall value above 0.3 in the predictions.

There are 146 records, corresponding to 146 employees of the company. Each employee or person has 21 features (financial + email +poi). We have 18 persons, reported as POIs. This gives us enough training examples to build a classifier. While mining the dataset, I came across data points like “TOTAL” and “THE TRAVEL AGENCY IN THE PARK” that contain missing or NaN values. So, I removed those records. Also, the record for “LOCKHART EUGENE E” contained only NaN values. Hence, they were removed too. Therefore, there were 143 records in the final dataset.

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”]

Ans : I ended up using ‘exercised_stock_options’, ‘bonus’, and ‘total_stock_value’ features in my ExtraTreesClassifier. To come to using these features , I used the “features_importances_” to find out the relative importance of all the features. The top three features and their relative importance are as follows:

bonus : 0.39902593

total_stock_value : 0.34827452
exercised_stock_options : 0.25269955
Also, I tried multiple combination of features and found, that the above three features gave me roughly 86% accuracy and a precision & recall value above 0.3. As I increased the features, the accuracy increased a bit, but precision and recall value started dropping. The scores that I obtained using the multiple combination of features are as follows:

Features	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**F2-Score**
total_stock_value,exercised_stock_options,bonus,deferred_income,restricted_stock,total_payments,salary,messages_from_poi_ratio,messages_to_poi_ratio	0.84867	0.34797	0.1545	0.21399	0.17383
total_stock_value,exercised_stock_options,bonus,deferred_income,restricted_stock,total_payments,salary,messages_from_poi_ratio,messages_to_poi_ratio,long_term_incentive, expenses	0.85413	0.37599	0.1425	0.20667	0.16271
total_stock_value,exercised_stock_options,bonus,deferred_income,restricted_stock,total_payments,salary,messages_from_poi_ratio,messages_to_poi_ratio,long_term_incentive, expenses,other, shared_receipt_with_poi	0.86067	0.4344	0.149	0.22189	0.17154
total_stock_value,exercised_stock_options,bonus,deferred_income,restricted_stock,total_payments,salary, messages_from_poi_ratio,messages_to_poi_ratio,long_term_incentive,expenses,other,shared_receipt_with_poi,loan_advances,director_fees,deferral_payments,restricted_stock_deferred,to_messages, from_messages	0.86247	0.45545	0.161	0.2379	0.18491
total_stock_value,exercised_stock_options,bonus,deferred_income,restricted_stock,total_payments,salary	0.85533	0.38062	0.1355	0.19985	0.15553
total_stock_value, exercised_stock_options, bonus, total_payments, salary, restricted_stock	0.86067	0.44289	0.1745	0.25036	0.19857
total_stock_value, exercised_stock_options, bonus, total_payments, salary	0.85847	0.4356	0.208	0.28156	0.23227
total_stock_value,exercised_stock_options,bonus,total_payments	0.86627	0.49744	0.291	0.36719	0.31734
total_stock_value, exercised_stock_options, bonus, salary, restricted_stock	0.84143	0.38798	0.1905	0.25553	0.21209
total_stock_value, exercised_stock_options, bonus, salary	0.83869	0.45305	0.234	0.30861	0.25905
total_stock_value, exercised_stock_options, bonus	0.86062	0.5864	0.319	0.41321	0.35101
total_stock_value, bonus, salary	0.82854	0.34794	0.131	0.19034	0.14966
bonus, total_stock_value	0.83069	0.42723	0.295	0.34901	0.31447
bonus	0.77744	0.49884	0.3235	0.39248	0.34796
total_stock_value	0.77154	0.23953	0.223	0.23097	0.22612

By keeping the above values and the following points in consideration, I came to decide upon the final features to be used :

1. The algorithm should give good recall value for it to be able to identify a person of interest correctly when there is. Hence, recall score is important, and should be greater than 0.3.
2. The algorithm should use minimum no. of features to avoid overfitting.
3. The algorithm should use the required features to capture the maximum trend in the data i.e it should have maximum training experience or maximal variance.

I wanted to figure out the communication style of persons of interest. Therefore, I added two features, 'messages_to_poi_ratio' and 'messages_from_poi_ratio' to check if they had any connection with an employee being a POI. But, I could not find any major impact these features had on improving the recall and accuracy score. So, they were not added to the final feature list.

I used ExtraTreesClassifier to build my model and did not use any feature scaling. This ensemble method differs from RandomForest classifier, in which by combining different decision trees (using subsets of training data and randomization) we come up with a more general solution and less overfitting. ExtraTreesClassifier is obtained by completey randomizing either the feature selection or the spilt threshold. This extra randomization gives more smooth decision boundaries in ExtraTreesClassifier. Also, it gives better performance because no bagging is used.

3. What algorithm did you end up using? What other one(s) did you try? How did model performance differ between algorithms? [relevant rubric term: "pick an algorithm"]

Ans: I worked with 4 different classifiers with the final feature list. The following statistics shows performance figures with the different approaches:

Final features list = ['bonus', 'total_stock_value', 'exercised_stock_options']

Algorithm	Accuracy	Precision	Recall
Naïve Bayes	0.843	0.48581	0.351
DecisionTreeClassifier	0.80285	0.36602	0.3845
RandomForestClassifier	0.86015	0.5901	0.298
ExtraTreesClassifier	0.86062	0.5864	0.319

As I worked my way up to ExtraTreesClassifier, Naive Bayes gave a fairly good accuracy of about 84% and precision & recall score also above 0.3. I could not tune it further, so I moved to Decision Tree which is more complicated model. I also checked the importance of features by feature_importances_. This model gave lower values than Naïve Bayes. I wanted to take a glimpse of the ensemble methods too. RandomForestClassifier gave better accuracy & precision score than Naïve bayes but lower recall score. And as I really want my model to correctly identify a POI when it appears. So recall score is important for my optimal model performance. I could not tune random forest any further. At last, I tried

ExtraTreesClassifier and achieved better figures than RandomForest. Hence, that is my final algorithm for this assignment.

4.What does it mean to tune a parameter 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 choice or a different model that does utilize parameter tuning, e.g. a decision tree classifier).[relevant rubric item: "tune the algorithm"]

Ans: Tuning a machine learning algorithm means to be able to set the parameters of algorithm to such optimal values that enable it to perform in the best way possible. To perform in the best way possible means, the algorithm should be able to capture the maximum trend in the data by having the best learning experience (neither underfit, nor overfit the data) and also translate its learning power to deliver the best accuracy, recall & precision scores during the prediction.

To get the optimal parameter values, I used GridSearchCV. It is a way to systematically work with multiple combination of parameter tunes, cross validating as it goes to determine which parameter tune gives the best performance. The beauty is that it can work through many combinations of parameters in only a couple extra lines of code.

Following the sklearn documentation for ExtraTreesClassifier, I used the following parameter values to use with GridSearchCV automated parameter tuning.

```
params = {'n_estimators':[1,5,10],'criterion':['gini','entropy'],'max_features':['sqrt','log2',None]}
```

```
The best parameter values were:  {'max_features': None, 'n_estimators': 5, 'criterion': 'entropy'}
The best parameter estimators were:
ExtraTreesClassifier(bootstrap=False, class_weight=None, criterion='entropy',
        max_depth=None, max_features=None, max_leaf_nodes=None,
        min_samples_leaf=1, min_samples_split=2,
        min_weight_fraction_leaf=0.0, n_estimators=5, n_jobs=1,
        oob_score=False, random_state=None, verbose=0, warm_start=False)
```

And the best recall score I could get was 0.39700. The corresponding precision and accuracy scores were 0.46029 and 0.83562 respectively.

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"]

Ans: Validation of a machine learning algorithm is basically to access whether the algorithm is doing what we want it to do. We can perform validation by splitting the data into training and testing sets. This mechanism gives estimate of performance on an independent testing dataset and also serves as a check on overfitting the training dataset. If we commit the mistake of training and testing on the same dataset, it will deliver the best accuracy, and will lead to an illusion that the model is performing amazingly well.

But, that is where we would be falling into the trap of overfitting the data. So, the best thing to do is hold out some part of the dataset for testing purpose.

I validated my analysis using the test_classifier function given in the starter code of the tester.py file. This function uses sklearn.cross_validation.StratifiedShuffleSplit() to do training-testing split. It is a blend of StratifiedKFold and ShuffleSplit, that returns stratified randomized folds. The folds are made by preserving the percentage of samples for each class. This mechanism is therefore suitable for this dataset as it randomizes the data while splitting, thereby giving balanced sets that contain POIs and Non POIs.

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"]

Ans: I used the following three metrics to evaluate the performance of my algorithm.

1. Accuracy: It is defined by the ratio of correct predictions to the total predictions. An accuracy score of 83% means, that 83% of the predictions made by the model as POIs are correct.
2. Precision: It is defined as how many selected items were relevant. It is ratio of no. of correctly predicted employees as POI to the total predictions made as POI. So, the precision score of 0.46029 means , we are 46% confident that an employee identified as POI is actually a POI.
3. Recall: It is defined as how many relevant items are selected. It is ratio of no. of correctly predicted employees as POI to the actual total no. of POIs. So, a recall score of 0.39700 means, that 39.7% of the times algorithm is able to correctly identify an employee as POI when it shows up in the dataset.