ENRON DATASET - ANALYSIS

***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?*

Question : Who are the people of Interest(POI's) in Enron?

DATASET:The Enron dataset has the records of email+financial data of a number of employees and also a POI label based on whether the official was indicted or not by the government

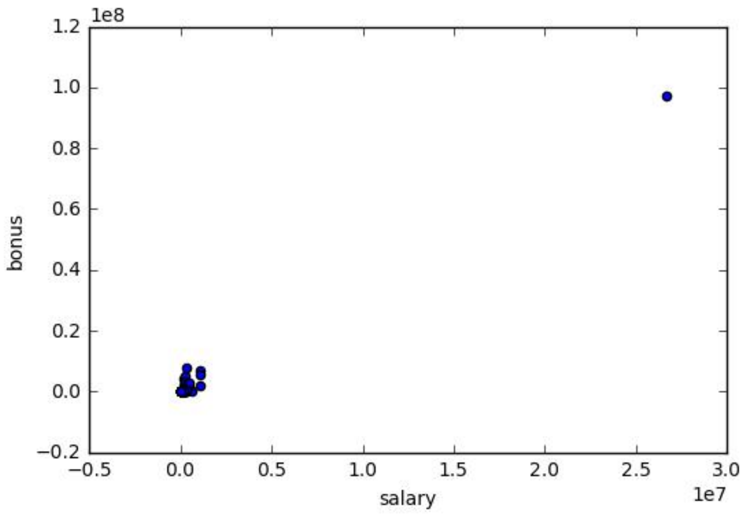
OBEJCTIVE: The objective is to come up with an algorithm applying the concepts of machine learning like feature selection/engineering and parameter tuning and evaluate multiple algorithms based on metrics to assess if the chosen algorithm does a good job of predicting the POI candidates.

EXPLORATORY ANALYSIS:
1.The dataset has 146 records.Details of 146 employees with their email and financial data
2.There are 21 features describing each of the records
3.There are 18 records which are labeled as POI
4.The features for which there are no data available are indicated as NaN and below is the count of number of records having NaNs against each of the feature

```
{'salary': 51, 'to_messages': 60, 'deferral_payments': 107, 'total_payments': 21, 'long_term_ince
ntive': 80, 'loan_advances': 142, 'bonus': 64, 'restricted_stock': 36, 'restricted_stock_deferred':
128, 'total_stock_value': 20, 'shared_receipt_with_poi': 60, 'from_poi_to_this_person': 60, 'exerc
ised_stock_options': 44, 'from_messages': 60, 'other': 53, 'from_this_person_to_poi': 60, 'deferr
ed_income': 97, 'expenses': 51, 'email_address': 35, 'director_fees': 129}
```

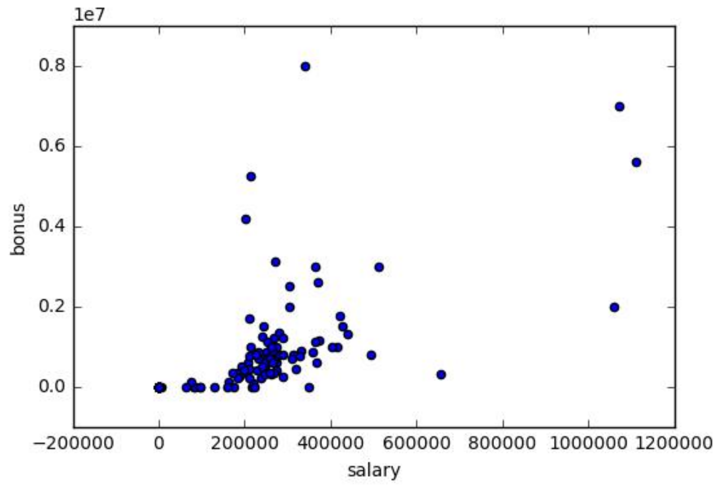
OUTLIERS:
We came across a couple of outliers which were removed
1.A record with no data for any of the features - REMOVED
2.A record which was a travel agency and not a person - REMOVED
3.Quite a handful of records which did not have any financial data - REMOVED

OTHER OUTLIER DETECTION AND REMOVAL/TREATMENT:
When mapping the bonus data and the salary data we came up with a record (point) in the visualization which was way larger than the mean of the rest of the data .



Using python code we were able to identify it as a record with the KEY = “TOTAL” which was the total of all the records and not pertaining any individual.
Hence the same has been removed from the dataset for further analysis.

After removing the TOTAL outlier plotted the data again.There were a couple of other records with way higher salary and bonus but they were not records to be removed but seemed like real POIs.



TOTAL PAYMENTS/total stock/expenses versus SALARY:
Analysing the total payments/expenses/total stock versus salary we came across some points which were way above the rest as above but cannot be considered outlier hence we are retaining the same for above reasons and will be included in analysis.

NAN handling for ‘Salary’ Entry:
There were 51 records which had NaN as an entry for salary. we have considered the median (since the mean would be skewed considering top people in the group)salary of the group and replaced it for the NaN values.This seemed more sensible than replacing it with a ‘0’.

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.

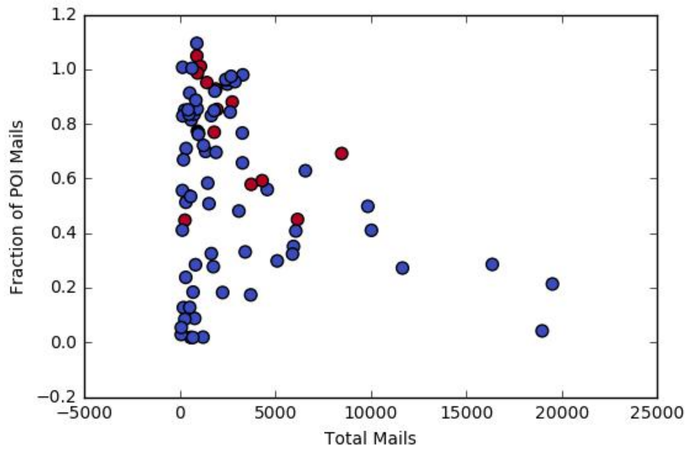
FEATURE ENGINEERING:
FRACTION OF POI MAILS
Using the e-mail data a new feature is developed which is the fraction of POI e-mail interaction versus total mail interaction.

Formula used is as below
POI_Interaction = from_this_person_to_poi+shared_receipt_with_poi+from_poi_to_this_person
Mail_Interaction = to_messages + from_messages

fraction_POI_interaction = POI_interaction/Mail_interaction

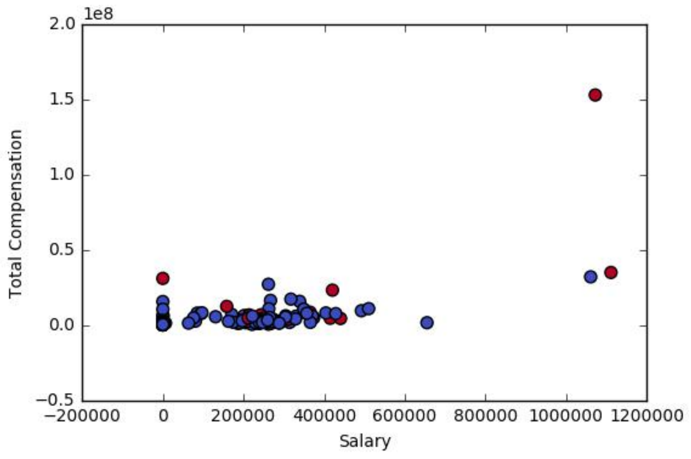
This feature just gives a value if the person had a lot of mail communication with POIs which inturn could turn the spotlight on him/her and give us a clue as to whether he/she is a POI.

However it was observed that the fraction had values greater than 1 (which is because of data errors) and we are imputing the same to 1 just as an indication of maximum communication with POIs.
From the below there are no clear clusters but can be seen that the fraction of POI is at least greater than 0.4 for those marked in red(POIs) .It can be safely considered that the ones who had very less interaction (<0.4) are most probably not in POI category



TOTAL COMPENSATION
A feature Total compensation is created which is the sum of total payments+ total stock value

In the below there are no clusters coming up which clearly indicate any distinction between POI and non POI



Both the above features are incorporated in the feature selection process and the algorithm is used to pick the best ones.

FEATURE SELECTION:
The existing features and the additionally engineered featured are fed into the ExtraTreesClassifier to get the feature_importances value for each of the feature and the same is used to trim the model to the relevant feature.

Choice of number of features:
A combination of Kbest and feature_importances from trees is used to arrive at the best features.

In the trees method adopted the median value of the importances are considered and those features with values greater than the median value is taken for further processing.The feature_ importances are dynamic and hence the number of features chosen also varies between 8-12.
To arrive at a constant selected featureset, multiple iterations of the above were performed and a union of the result along with the result yielded from Kbest was considered for the final list of 15 features.
Some intuition has also gone into the selection of the final list.Like the number of from_mails/to_mails cannot determine a person as a POI.The records having valid entries for restricted_stock_deferred or for deferral_payments are very less and hence can add negligible value to the algorithm.The ones marked in red have been removed.

Features	KBest Scores	Feature_imp-iter 1	Feature_imp-iter 2	Feature_imp-iter 3
Salary	7.57E+00	0.05855429	0	0.09498808
deferral_payments	6.92E-01	0	0	0
total_payments	5.13E+00	0.08315694	0.08315694	0.08315694
loan_advances	5.01E+00	0	0	0
bonus	1.12E+01	0.1113461	0	0.06580387
restricted_stock_deferred	1.06E-01	0	0	0
deferred_income	6.66E+00	0	0	0
total_stock_value	1.47E+01	0.14258152	0.14258152	0.14258152
expenses	3.19E+00	0.05735931	0	0
exercised_stock_options	1.54E+01	0.19172113	0.21681583	0.19172113
other	2.11E+00	0	0.0334596	0
long_term_incentive	4.56E+00	0	0.03643378	0
restricted_stock	5.03E+00	0.05890201	0.05890201	0.05890201
to_messages	3.43E-01	0.11291286	0.09021055	0.05735931
from_poi_to_this_person	2.10E+00	0	0.04554223	0
from_messages	4.63E-01	0	0	0.04554223
from_this_person_to_poi	9.44E-01	0.04774274	0.04774274	0.04774274
shared_receipt_with_poi	3.68E+00	0	0.06580387	0.07647908
fraction_POI_interaction	5.52E+00	0.10287186	0.10287186	0.10287186
Mail_interaction	8.27E-03	0.03285124	0.07647908	0.03285124

CHOICE OF FINAL FEATURES

```
'salary',
'total_payments',
'loan_advances',
'bonus',
'deferred_income',
'total_stock_value',
'expenses',
'exercised_stock_options',
'long_term_incentive',
'other',
'restricted_stock',
'from_poi_to_this_person',
'from_this_person_to_poi',
'shared_receipt_with_poi',
'fraction_POI_interaction'
```

FEATURE SCALING:
Since a lot of features are of a different scale feature scaling using the minmax scaler is used for normalizing the same before feeding into the algorithm.

3.What algorithm did you end up using? What other one(s) did you try? How did model performance differ between algorithms?
Below are the 4 algorithms chosen for trial purposes.

- ✓ Naive Bayes
- ✓ SVC
- ✓ Decision Tree
- ✓ Random Forest

Below is the model performance in terms of accuracy scores.

After splitting the data into training and test set scores as below with default parameters.

```
Accuracy Score of Naive Bayes: 0.84375
Accuracy Score of SVC: 0.9375.
Accuracy Score of Tree: 0.8125
Accuracy Score of Random Forest: 0.84375
```

Final Choice :
The data set is small and the samples are imbalanced(number of POI's is 18 for a datasize of 146) Hence accuracy score is not the best indicator of the algorithm.Even if the algorithm ends up predicting all of them as non POIs then the accuracy score would be 87%.Hence different metrics has been addressed in the choice of algorithms

The final choice of algorithm was Decision Trees after tuning.
Steps explained below

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).

From the project go to know how well tuning can affect the performance.
Explaining the process in my own understanding:
Every algorithm has a set of default values using which the model if fitted and prediction is carried out.Tuning is about setting the right parameters overriding the default parameters in accordance with the requirement of the objective and also ensure that the algorithm is not overly-fitted.

For Decision Trees the parameters were tuned using the param_grid of the GridsearchCv method. The GridSearchCv helps in trying out various hyper parameters combination and picks the bext combination of parameters which is then used to fit the data and for prediction.

Best Parameters:

```
{'DT__class_weight': 'balanced', 'DT__max_features': 'auto', 'DT__max_depth': 3, 'DT__min_samp
les_leaf': 5, 'DT__criterion': 'gini', 'DT__random_state': 42}
```

For algorithms which do not have parameters:

Certain algorithms like Naive Bayes - Gaussian NB do not have parameter tuning.It can be noted that feeding the algorithm with highly co-related features can add more weights to the same and bring down the performance.Hence some of the highly co-related features were pruned and a subset of the final feature list was passed to this algorithm.Also PCA was used for dimension reduction.Pipelines were used to mention the steps.
Pruning was done outside of pipeline and the steps involved PCA followed by scaling followed by algorithm fit.

5.What is validation, and what's a classic mistake you can make if you do it wrong? How did you validate your analysis?

Validation is a process of testing the algorithm in unseen data and assessing the performance on this new data.

Mistake: Overfitting can arise if the algorithm is not properly validated.Classic mistake it to train and test the algorithm using the same data.The algorithm has fine tuned its parameters to fit the given data.If the same data is used to test the algorithm and fine tune it   then the algorithm ends up memorizing the given data and performs very poorly on unseen data

Validation:Before tuning the algorithm was tested on separate training and test sets.
At the tuning step a cross validation method called stratifiedshuffleplit is used(refer to the GridSearchCv step) which is a combination of Stratified Kfold and shufflesplit and returns test and train indexes from the entire data.

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.

For this project 2 evaluation metrices were closely monitored to tune and end up with the final choice.

1.Precision
2.Recall

Output from poi_id.py

```
Accuracy  Score  :  0.716981132075
Precision:  0.807449494949
Recall:  0.676449275362
F1score:  0.664273648649
```

Output of tester.py

```
Accuracy:  0.67155     Precision:  0.31994     Recall:  0.71650        F1:  0.44235            F2:  0.5741
6
Total  predictions:  11000
True  positives:  1433
False  positives:  3046
False  negatives:    567
True  negatives:  5954
```

Interpretation of the metrics:
Precision Score: 0.3199.
True positives are the POI which were identified correctly by the algorithm
False positives are normal people who were classified as POIs by the algorithm.

Among the number of people identified as POIs by the algorithm what is the percentage of people who were actually POIs
  Precision = 31.99%
If the algorithm picks 100 people as POIs then 31 of them will be POIs

Recall Score: 0.716
True positives   are the POI which were identified correctly by the algorithm
False Positives are POI who were not identified as POIs by the algorithms.

Recall score gives us the percentage of how many POIs the algorithm was able to correctly identify.
= 71.6%
It means the algorithm can identify 71 among 100 POIs in a given set.