Detection of Fraud in Enron Data

In 2000, Enron was one of the largest companies in the United States. By 2002, it had collapsed into bankruptcy due to widespread corporate fraud. In the resulting Federal investigation, a significant amount of typically confidential information entered into the public record, including tens of thousands of emails and detailed financial data for top executives.

The goal of this project is to identify persons of interest (POI) defined as people from the Enron dataset, who were involved in corporate fraud from the financial and email data.

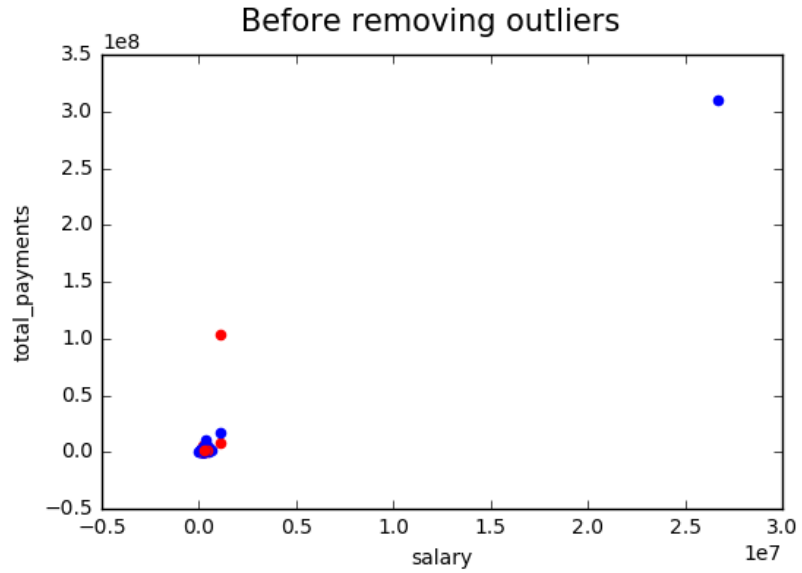
Data Exploration:

The dataset contains 146 data points of whom 18 are persons of interest. There are 21 features for each person in the data set: 'salary', 'to_messages', 'deferral_payments','total_payments','exercised_stock_options','bonus','restricted_stock','shared_receipt_with_poi','restricted_stock_deferred','total_stock_value','expenses','loan_advances','from_messages','other','from_this_person_to_poi','poi','director_fees','deferred_income','long_term_incentive','email_address','from_poi_to_this_person'.

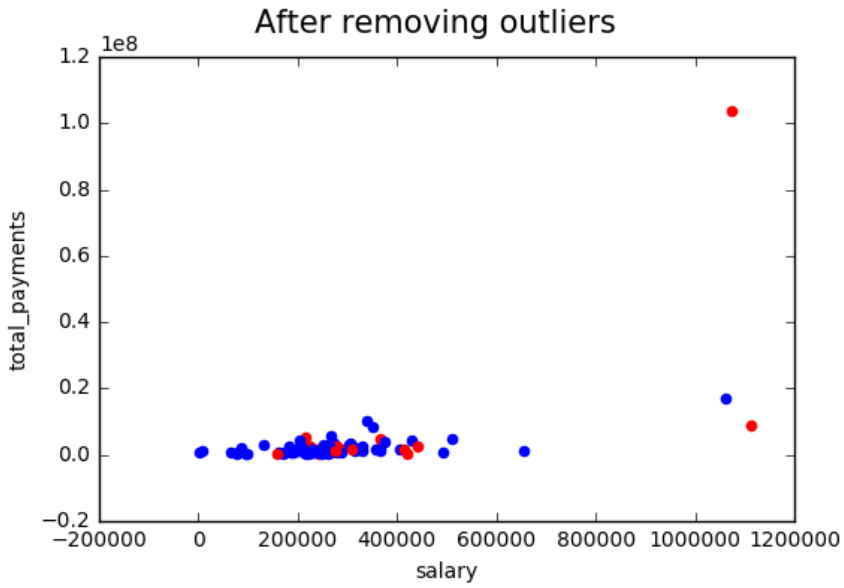
Among features with large missing values greater than 100 are loan_advances, director_fees, restricted_stock_deferred and deferral payments. The features selected for this report are and their corresponding no. of missing values are: 'poi'-0,'deferred_income'-97,'bonus'-64,'fraction_total_messages_poi'-60,'shared_receipt_with_poi'-60

Outlier Detection:

Plotting a few features a few outliers show up which were isolated.

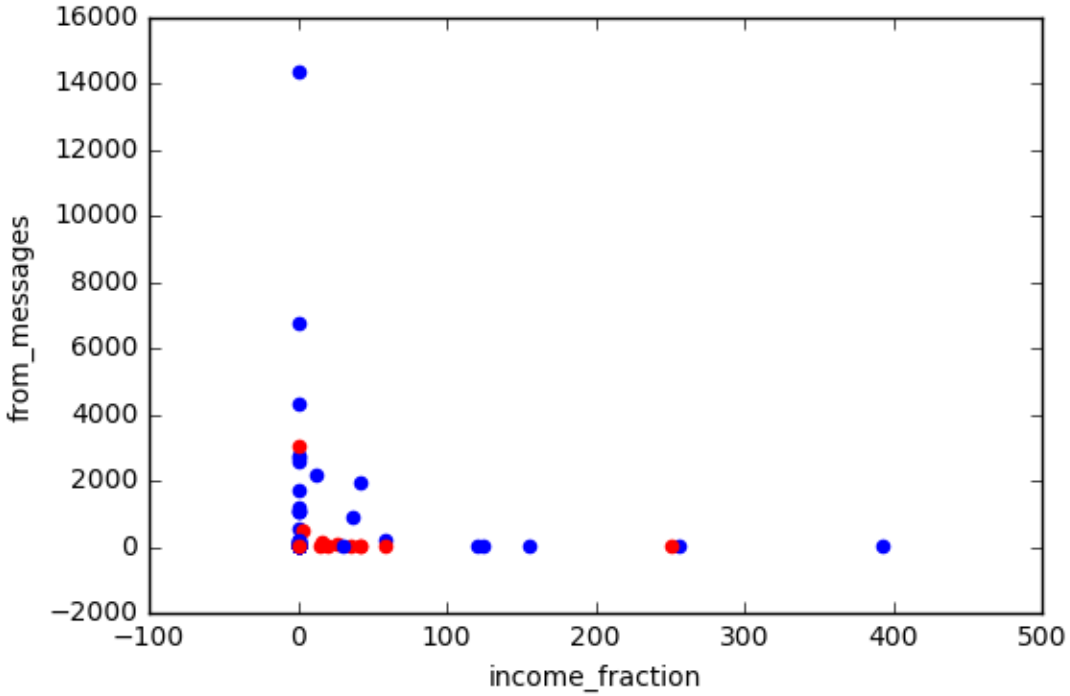


Outliers were isolated visually to see if they need to be taken out before going into further an alysis to remove any bias in the data. The outliers seen from the plot are ‘TOTAL’ and ‘LAY KENNETH L’. ‘TOTAL’ is column totals of the data and since this is not a real data point it is removed. Other outliers are actual POI’s and information in line with the purpose of data coll ection for this report that would go into the prediction model so they are not removed. The n ewly plotted data is as below after removing outlier ‘TOTAL’.



Feature selection and engineering:

Decision tree algorithm is used in feature selection. Feature scaling was not performed as the algorithm doesn’t require it. The top four features were selected for use in algorithm selection. A new feature was also engineered called ‘income_fraction’ which equals (–deferred_income/1000+expenses)/shared_receipt_with_poi. Several features were attempted and plotted to visually interpret if they achieve separation between poi and non-poi. Income_fraction seemed to achieve a fair bit of homogenous clustering of poi compared to non poi when plotted, as seen below.



The feature importances as given by the Decision Tree classifier are:

Feature	Feature importance
expenses	0.243008314437
income_fraction	0.207373271889
exercised_stock_options	0.123314558798
fraction_total_messages_poi	0.10582010582
restricted_stock_deferred	0.10582010582
from_this_person_to_poi	0.0617964903679
restricted_stock	0.047619047619
fraction_shared_receipt_poi	0.042328042328
total_stock_value	0.031746031746
deferred_income	0.031174031174

Feature selection:

The combination of features selected for final analysis were the ones which gave the maximum accuracy and precision and recall scores.

Decision tree classifier scores:

Feature list	Accuracy	Precision	Recall
['poi','to_messages','deferral_payments','expenses','deferred_income','long_term_incentive','shared_receipt_with_poi','loan_advances','from_messages','director_fees', 'bonus','total_stock_value','from_poi_to_this_person','from_this_person_to_poi','fraction_total_messages_poi', 'restricted_stock','salary','total_payments','exercised_stock_options','income_fraction']	0.8340	0.335	0.246
['poi','restricted_stock','total_stock_value','fraction_total_messages_poi','exercised_stock_options','from_this_person_to_poi','deferred_income','income_fraction']	0.838	0.39	0.23
['poi','total_stock_value','exercised_stock_options','from_this_person_to_poi','deferred_income','fraction_total_messages_poi','income_fraction']	0.839	0.393	0.231
['poi','exercised_stock_options','fraction_total_messages_poi','deferred_income','from_this_person_to_poi','income_fraction']	0.841	0.40	0.225
['poi','fraction_total_messages_poi','deferred_income','from_this_person_to_poi','income_fraction']	0.884	0.407	0.353

Naïve Bayes scores

Feature list	Accuracy	Precision	Recall
['poi','to_messages','deferral_payments','expenses','deferred_income','long_term_incentive','shared_receipt_with_poi','loan_advances','from_messages','director_fees', 'bonus','total_stock_value','from_poi_to_this_person','from_this_person_to_poi','fraction_total_messages_poi', 'restricted_stock','salary','total_payments','exercised_stock_options','income_fraction']	0.78	0.228	0.28
['poi','restricted_stock','total_stock_value','fraction_total_messages_poi','exercised_stock_options','from_this_person_to_poi','deferred_income','income_fraction']	0.862	0.527	0.38

['poi','total_stock_value','exercised_stock_options','from_this_person_to_poi','deferred_income','fraction_total_messages_poi','income_fraction']	0.8601	0.514	0.38
['poi','exercised_stock_options','fraction_total_messages_poi','deferred_income','from_this_person_to_poi','income_fraction']	0.872	0.581	0.373
['poi','fraction_total_messages_poi','deferred_income','from_this_person_to_poi','income_fraction']	0.846	0.210	0.193

Algorithm Tuning

Naïve Bayes algorithm was attempted with default parameters to check the accuracy, which resulted in 0.84 accuracy. Decision Tree classifier returned an accuracy of 0.77 prior to tuning the parameters and Support Vector Machine classifier along with feature transformation using PCA returned an accuracy of 0.73.

Prediction power of the new feature:

Naïve Bayes

Score	**Without feature**	**With feature**
Accuracy	0.872	0.872
Precision	0.5814	0.581
Recall	0.375	0.375

Decision Tree classifier (before tuning)

Score	**Without feature**	**With feature**
Accuracy	0.812	0.822
Precision	0.162	0.246
Recall	0.211	0.376

Decision classifier tuning parameters

Min_samples_split	**Max_depth**	**Min_samples_leaf**	**Precision**	**Recall**
2	2	1	0.388	0.437
2	2	2	0.402	0.44
2	2	3	0.400	0.44
2	3	1	0.41	0.354
2	**3**	**2**	**0.474**	**0.354**
2	3	3	0.467	0.354
2	3	4	0.464	0.356
2	4	2	0.366	0.323

SVM did not yield any true positive or true negatives to calculate the accuracy scores.

Parameter tuning of algorithms gives the optimal conditions to set the algorithm so we can get the best accuracy. This ensures that the dataset you have is well understood by the algorithm that it is being modeled on. Although choosing the right algorithm is up to us once you decide on which algorithm to use, parameter tuning is essential so we are using the right conditions to achieve the optimal accuracy. The final algorithm that was chosen here, Naïve Bayes classifier, did not have any parameters to tune. The accuracy, precision and recall achieved are 0.87, 0.581 and 0.375. Decision Tree was tuned using GridSearchCV on parameters max_samples_split and max_depth and min_samples_leaf. To optimize on precision and recall instead of accuracy scores, manual tuning was performed to arrive at optimal parameters [max_samples_split:2,max_depth:3,min_samples_leaf:2]. Support Vector Machine and PCA were also attempted to find the optimal parameters via GridSearchCV but since algorithm did not return any true positives or true negatives. Considering the number of features (18) and the size of the dataset available after removing NA's, SVM is not expected to perform well, which is what was observed.

Validation and evaluation:

Precision and recall are used to check the validity of the algorithm performance. Precision is the probability that of all the instances that are classified as POI, the classification is actually, true, that is the predicted instance is actually a POI. Recall is the probability a POI will be classified as POI or a non-POI is classified as non-POI. The final algorithm chosen here gives a precision and recall of 0.534 and 0.34 respectively.

Validation is important so as to check how the model performs on new data sets, to see if the results are replicable and the data is not overfitted. Splitting the data into test and train sets helps optimize the data available for training the algorithm, and prevent overfitting when the train data is too small. The algorithm gives a high accuracy score on train but might not perform so well on the test. KFold validation is used to split the data into training and testing sets. The optimal performance of the algorithm is achieved at n=2 folds.