Identifying Fraud in Enron Emails

Introduction

This project is an investigation into the Enron fraud case. Due to the Federal investigation, a large amount of e-mails surrounding the Enron employees became public. Because of the sheer volume of information that was released, it becomes very time consuming to review each one individually. Utilizing machine learning, the information can be quickly processed to identify certain people or features that may be of interest. Different machine learning algorithms will be tested to accurately identify persons of interest within the dataset.

Data Exploration

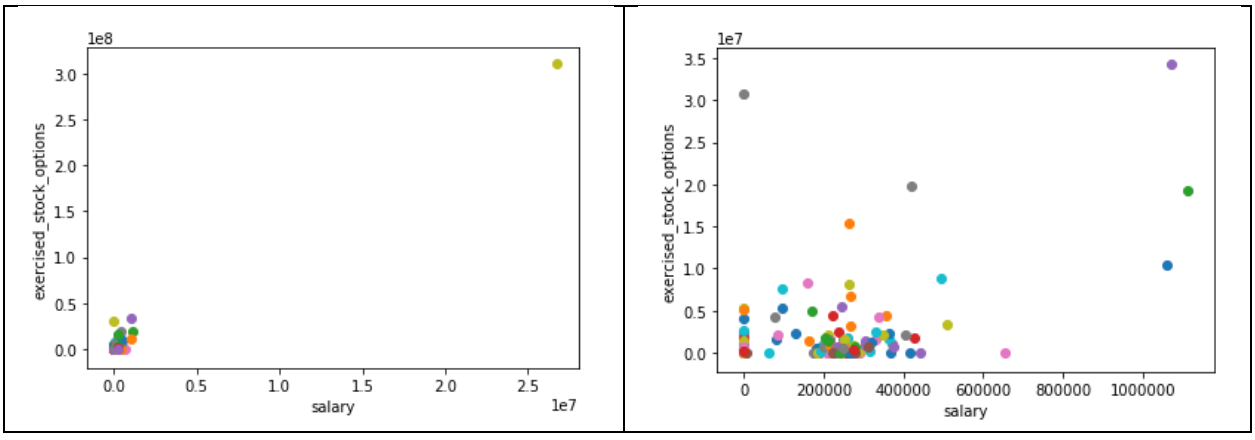
Initial Review

During the initial review of the dataset, there appears to be 146 employees, with 20 different features. 14 of these features are financial information, and 6 features are e-mail information. One feature, "email address" is being disregarded as it provides no meaningful information for this analysis. 18 people are identified as "persons of interest", making up 12.33% of the total employees.

All features are missing information for at least one employee. 15 features are missing information for at least 50 employees, and 4 features are missing information for over 100 employees. None of these features will be removed at this time, as the presence of some of these values could indicate a pattern to identify a person of interest.

Outliers

When doing an initial plot of salary and exercised stock options, the left plot was generated:



This plot shows an extreme outlier, which needed to be investigated. The max value and person for salary was identified as: 26704229, TOTAL. This indicates that one of the "people" in the dataset contains the total information for the sum of values. After the outlier was removed from the dataset, the plot on the right was generated, which shows a more realistic distribution.

One user, “LOCKHART EUGENE E” was missing data for all features, and was removed from the dataset. Another use, “THE TRAVEL AGENCY IN THE PARK” was also removed, as the name indicates that it is not a person.

Identifying Persons of Interest

Feature Selection

Three features were selected to be used in identifying Persons of Interest: exercised_stock_options, total_stock_value, and bonus.

New Features

Three features were added to the dataset: bonus_salary_ratio, emails_to_poi_ratio, and emails_from_poi_ratio. The e-mail features were created because people can send and receive varying amounts of emails. A person who sends out 3000 emails but only 15 to a person of interest would have a higher “emails to poi” count than someone who sends out 50 emails, 10 of which going to a person of interest. These ratios are to gauge what percent of an employee’s email communication is with a person of interest. Additionally, the bonus_salary_ratio was created because it appeared that some persons of interest received huge bonuses in relation to their salary, and this could potentially lead to identifying other persons of interest.

Evaluating Features

The three best features were selected by utilizing the scikit-learn library. Using the SelectKBest function, the ANOVA F-value was calculated for each feature. The features with the highest scores were then selected for use in identifying our Persons of Interest. Below is a list of each feature and their score:

Exercised_stock_options	24.8151
Total_stock_value	24.1829
Bonus	20.7923
Salary	18.2897
Emails_to_poi_ratio	16.4097
Deferred_income	11.4585
Bonus_salary_ratio	10.7836
Long_term_incentive	9.9222
Restricted_stock	9.2128
Total_payments	8.7728
Shared_receipt_with_poi	8.5894
Loan_advances	7.1841
Expenses	6.0942
From_poi_to_this_person	5.2434
Other	4.1875
Emails_from_poi_ratio	3.1281
From_this_person_to_poi	2.3826
Director_fees	2.1263
To_messages	1.6463
Deferral_payments	0.2246
From_messages	0.1697
Restricted_stock_deferred	0.0655

In the above table, the highlighted cells represent the new features that were engineered for this analysis. It is interesting to note that Bonus and Salary are significant in identifying persons of interest, yet the "bonus to salary ratio" is almost half as effective as a feature. Additionally, the "emails to poi ratio" is fairly significant, although "from poi to this person" and "from messages" are not impactful at all. This indicates that the number of messages sent do not matter, but the percentage of the messages being sent to a person of interest is.

Number of Features

To determine the best number of features to use, two to ten features were tested against different machine learning algorithms. Below is a table with the results:

	Logistic Regression		SVM		Naïve Bayes		AdaBoost		Decision Tree	
Features	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
2	0.7113	0.2045	0.34731	0.087	0.46889	0.2675	0.54791	0.243	0.46684	0.176
3	0.65613	0.166	0.63683	0.1245	0.48581	0.35100	0.39689	0.332	0.40683	0.31550
4	0.61224	0.15	0.68718	0.067	0.50312	0.323	0.24164	0.206	0.4041	0.276
5	0.48153	0.1955	0.24976	0.2605	0.45558	0.3	0.2996	0.259	0.29675	0.2325
6	0.55351	0.225	0.39687	0.076	0.47723	0.351	0.34598	0.2325	0.25266	0.214
7	0.53106	0.2265	0.37739	0.1285	0.44189	0.2985	0.37171	0.2825	0.29795	0.2035
8	0.52375	0.226	0.39226	0.152	0.42857	0.312	0.15243	0.24	0.36736	0.2285
9	0.52281	0.2235	0.27455	0.137	0.41129	0.306	0.25363	0.1835	0.36889	0.23
10	0.45292	0.19	0.22472	0.11	0.33982	0.262	0.29388	0.221	0.28144	0.2495

As shown above, it appears that using 3 features will provide the best results, so the 3 best features will be used to make the predictions. The cells that are highlighted represent the features and algorithms where both precision and recall are greater than .3.

The three features selected are the ones with the highest ANOVA F-score, which are exercised_stock_options ,total_stock_value, and bonus.

Algorithm Selection

Five algorithms were selected to identify Persons of Interest: Naïve Bayes, Support Vector Machines, Decision Trees, Ada Boost, and Logistic Regression.

Preprocessing

To improve the performance of some of the algorithms, the feature information is scaled using the MinMaxScaler. Because values such as bonus, which can range from 0 to 8,000,000 have such a large range, transforming the data into a simpler scale will decrease the processing time for the algorithms. The Support Vector Machine algorithm largely benefits from scaling the data prior to using the algorithm.

Parameter Tuning

To determine the best algorithm for this analysis, the parameters of the Support Vector Machine, Decision Tree, Ada Boost, and Logistic Regression algorithms were tuned. Naïve Bayes requires no

parameters. Tuning the parameters is necessary to ensure the algorithm fits the data without overfitting it. Overfitting the data leads to a decrease in performance of the algorithm, and properly tuning the parameters ensures the highest performance.

To determine the optimal parameters for the algorithms, GridSearchCV was utilized. For each algorithm, a range of parameter values were inputted, and then tested to determine which parameter values produced the highest score. As we want to test for the highest precision and recall scores, the F1 score was utilized for the ranking, as it is the harmonic mean of precision and recall.

For example, for the Ada Boost algorithm, various values of "n_estimators", "learning rate" and "algorithm" were tested. The results of GridSearchCV returned the following values as the best parameter values:

{'n_estimators': 40, 'learning_rate': 1.8, 'algorithm': 'SAMME'}

The Ada Boost algorithm was then tuned with these parameters, and then tested. Unfortunately, the algorithm did not perform as well as expected, as the Precision was 0.34598 and Recall was 0.23250.

Choice

The algorithm that performed the best at identifying Persons of Interest is the Naïve Bayes algorithm. This was particularly interesting, as this algorithm has no parameters to tune.

Validation

Validation is the process in which a trained algorithm is assessed using testing data. This is necessary to judge the performance of the algorithm. To perform validation, the data is split into two groups: training and testing data. The algorithm is trained utilizing the training data, and then it is validated by using the testing data. Because the testing data is not used for training, the algorithm must rely on the decision boundaries established from the training data to make predictions on the testing data. After performing validation on multiple algorithms with different parameters, the best algorithm can be selected and used on new data.

The performance of each algorithm can be measured using a variety of metrics, and it is important to select the best performance metrics for the dataset. Accuracy score, precision, recall, and f1 scores are the most common performance metrics, however not all of them may be appropriate measures of performance.

Performance Metrics

Algorithm performance was measured by the precision and recall scores. While accuracy could have been utilized to measure the algorithm performance, it was not selected because the task involved identifying a small number of people. There were instances in testing where accuracy was over 85%, however precision and recall were both 0. This was because the algorithm flagged every user as a non-Person of Interest. As we want to identify the persons of interest, the accuracy score proved to not be a good indicator of algorithm performance.

Precision is a measure of how many persons identified as a persons of interest, are in fact persons of interest. Recall is a measure of how many persons of interest were correctly identified as a person of interest. This is why precision and recall are the best performance metrics to compare algorithm performance.

The precision and recall scores utilize the following formulas:

$$Precision = \frac{true\ positives}{true\ positives + false\ positives}$$

$$Recall = \frac{true\ positives}{true\ positives + false\ negatives}$$

Cross Validation

To test the performance of the algorithms, there needed to be cross validation. Using a training/test split was considered, however due to the small number of persons of interest within the dataset (12.33%), there was a possibility for there to not be enough persons of interest present in the training or testing data sets for the algorithm to properly learn and identify.

The StratifiedShuffleSplit was utilized for testing, as it does the training/test split but can perform it multiple times. For each algorithm, this process was repeated 1000 times.

Algorithm Performance

The performance of each algorithm (with optimized parameter tuning) using 3 features is in the table below:

	Naïve Bayes	SVM	AdaBoost	Decision Tree	Logistic Reg.
Precision	0.48581	0.63683	0.39689	0.40683	0.65613
Recall	0.35100	0.1245	0.332	0.31550	0.166

As shown in the table above, the Naïve Bayes, AdaBoost, and Decision Tree algorithms performed the best, with both having Precision and Recall scores greater than .3. Between the three methods, Naïve Bayes has the highest performance in both precision and recall.

For the Naïve Bayes algorithm, 47.723% of persons it identified as persons of interest were in fact persons of interest. Additionally, 35.1% of persons of interest were correctly identified as persons of interest.

Conclusion

While machine learning can process and classify information very quickly, it is important that this processing be done accurately. The Enron dataset was analyzed, had outliers removed, and new features were added that could increase the performance of machine learning algorithms. Several algorithms were optimized and tested, however even the best performing one still had questionable accuracy. The Naïve Bayes algorithm failed to correctly identify a person of interest 64.9% of the time, and 52.3% of persons of interest identified were not actually persons of interest.

This could be attributed to the limited size of the dataset. With only 146 data points, three of which were removed, there was not enough data for the algorithm to be able to create proper decision boundaries.