Project 5: Identify Fraud from Enron Email

Project Overview

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. In this project, you will play detective, and put your new skills to use by building a person of interest identifier based on financial and email data made public because of the Enron scandal. To assist you in your detective work, we've combined this data with a hand-generated list of persons of interest in the fraud case, which means individuals who were indicted, reached a settlement or plea deal with the government or testified in exchange for prosecution immunity.

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 goal of the project is to identify employees i.e. persons of interest (person of interest (POI) as an individual who was indicted, reached a settlement or plea deal with the government, or testified in exchange for prosecution immunity) from Enron who may have committed fraud based on the public Enron financial and email dataset.

Machine learning is the science of designing and applying algorithms that are able to learn things from past cases. It uses complex algorithms that iterate over large data sets and analyze the patterns in data. Machine learning is going to be very useful in these case as it will help us to rectify the patterns and recognize POI.

Talking about the background of the dataset, There were 146 Enron employees out of which 18 of them are POIs. There are 21 features.

Based on the Exploratory Data Analysis, 3 records were found which needed to be removed were TOTAL, THE TRAVEL AGENCY IN THE PARK and LOCKHART EUGENE E.

What features did you end up using 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 importance 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”]

Two more features were created which are poi_communication and total_wealth. poi_communication is a fraction of emails related to POI which would help a proportion of messages communicated POI to the total messages. total_wealth is the sum of values of salary, bonus, total stock value, exercised stock options.

Based on SelectKBest, the top 10 features are:

Features	Scores
Exercised Stock Options	24.8
Total Stock Value	24.1
Bonus	20.7

Salary	18.2
Total Wealth	15.3
Deferred Income	11.4
Long-Term Incentive	9.9
Restricted Stock	9.2
Total Payments	8.7
Shared Receipt with POI	8.5

The new feature **Total Wealth** appears in the above top 10 best features. So we can include it in the final list of important features.

The univariate feature selection process, select k-best was used in a pipeline with grid search to select the features. Select k-best removes all but the k highest scoring features. The number of features, 'k', was chosen through an exhaustive grid search, intending to maximize precision and recall values**.**

Features Scaling**:** It is crucial because it standardizes range of features in the data. I have used the StandardScaler() function in Scikit-learn standardize the features in the dataset.

Parameters Tuning is changing the algorithm input parameters which helps to improve the performance of algorithms.

For this, I have used Pipeline function, where parameters are tuned using StratifiedShuffleSplit and GridSearchCV.

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

I choose 3 algorithms and tuned the parameters with GridSearchCV to get the best precision and recall results. The algorithms are:

1. Logistic Regression
2. Support Vector Machine
3. Decision Tree

Among all these algorithms, Support vector machines gave the best accuracy. when used the right parameters to tune the models, the performance of these algorithms improved significantly.

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). [relevant rubric item: "tune the algorithm"]

Tuning the parameters of an algorithm means adjusting the parameters in a certain way to achieve optimal algorithm performance. It is about tuning those parameters to so optimal values that enable you to complete a learning task in the best way possible. Thus, tuning an algorithm or machine learning technique, can be simply thought of as process which one goes through in which they optimize the parameters that impact the model in order to enable the algorithm to perform the best. The parameters that I used to tune the parameters are:

1. Logistic Regression (LogisticRegression) : C, penalty, random_state
2. Support Vector Classifier(SVC) : C, gamma, kernel
3. DecisionTree:min_samples_split, min_samples_leaf, criterion, max_depth, r andom_state

I played around with the number of features that gave the optimal precision. I started inputting around 15 features but then when I decreased the number of features to half around 7 to 8 the performance of the models was much better.

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"]

Through Validation, one can substantiate the performance of a machine learning algorithm. It also prevents overfitting, It is important to break your datasets into training and testing sets.

To perform on autonomous datasets and battle overfitting, sklearn has a few assistant capacities under Cross-validation. The one utilized as a part of this undertaking is Stratifiedshuffle splits, which is especially appropriate for this dataset. StratifiedShuffleSplit guarantees that the percentage of target labels is roughly same in both training and validation set as it is in the entire dataset.

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"]

We use accuracy commonly to measure the performance of our algorithm but we know that it is not the only form of correct measurement at times. The other important metrics which can be used are precision and recall.

- **True positive**: when a label is a POI and the algorithm predicts it is a POI.
- **False positive**: when a label *is not* a POI and the algorithm predicts it is a POI.
- **True negative**: when a label is not a POI and the algorithm predicts a non-POI.
- **False negative**: when a label is not a POI and the algorithm predicts a POI.

Precision: Out of all items that are truly positive, how many are correctly classified as positive. It is calculated as (TP)/(TP + FP).

Recall: Out of all items are predicted as positive, how many are truly belong to the positive case. It is calculated as (TP)/(TP + FN).

Algorithm	Precision	Recall
Logistic Regression	0.45	0.40
Decision Tree	0.19	0.16
Support Vector Machine	0.43	0.13

The precision for logistic regression is the probability that a man who is distinguished as a POI is really a genuine POI; the way this is 0.45 implies that utilizing this identifier to signal POI's would bring about 55% of the positive flags being a false alarm. Recall measures are the probability that identifier will signal a POI in the test set. 40% of the time it would get that individual, and 60% of the time it wouldn't.

So, The logistic regression gives the best overall scores for the evaluation metrics.