Enron Email Data Machine Learning Project

Introduction

The goal of this project was to create a person of interest classifier using Enron data. Machine learning can be used to predict whether an employee is a person of interest based on their data. The Enron data contains employee financial (salary, bonus etc.) and email information (number of emails to and from a person of interest, etc). Persons of interest (those who were indicted, reached a settlement or testified) were identified in the dataset.

Data Exploration

There were a total of 146 data points in the set, with 18 employees identified as persons of interest. There were 19 features in the dataset (including all financial and email information except email address). Three values in the dataset were removed ('TOTAL', 'THE TRAVEL AGENCY BY THE PARK' and 'LOCKHART EUGENE E'). 'TOTAL' contains the sum values of the spreadsheet data and 'THE TRAVEL AGENCY BY THE PARK' is an account associated with travel expenses (as shown in the insiderpay.pdf document). 'LOCKHART EUGENE E' only has 'NaN' values for all entries as shown in the insiderpay.pdf document and confirmed from the data dictionary entry. The 'NaN' values in the dataset were retained as this seems to represent that a feature is not applicable to a person as shown in the insiderpay.pdf document. Many features contained 'NaN' values including 'restricted_stock_deferred' (126 NaN's) and 'deferral_payments' (105 NaN's). Additional outliers were not removed since only 18 persons of interest are present out of 146 data points.

Feature Selection and Creation

Features were scaled using MinMaxScaler before feature selection was implemented. This was due to the difference in magnitude of the features. For example, total_payments ranged from 0 to 103,559,800 while from_this_person_to_poi ranged from 0 to 528.

A new feature call 'total' was created which combined the values of salary, bonus and total_stock_value. This feature was created under the assumption that persons of interest were motivated by these financial rewards to perpetuate the fraud. SelectKBest was used in the pipeline to select features. The 5 selected features and scores were salary (18.3), bonus (20.8), total_stock_value (24.2), exercised_stock_options (24.8) and total (29.0). The new feature 'total' was included in the features selected by SelectKBest and had the highest score (29.0). This set of features was chosen using GridSearchCV including a pipeline with MinMaxScaler and the DecisionTree classifier. The SelectKBest k parameter in the pipeline was set in the range from 1 to 15 to include most of the features in the dataset.

Classifier Selection and Performance

The Decision Tree classifier had the best accuracy and recall compared with the Gaussian Naive Bayes and Support Vector classifiers as shown in Table 1 below, while the Support Vector Classifier had the best recall score. The Decision Tree classifier was used for the final algorithm. Features were scaled using MinMaxScaler and features were selected using SelectKBest for all classifiers.

Table 1 – Classifier performance

	Accuracy	Precision	Recall
Gaussian NB	0.85	0.41	0.33
SVC	0.72	0.27	0.63
Decision Tree	0.87	0.51	0.46

Parameter Tuning

Tuning an algorithm's parameter means finding the parameters which allow for the best performance of the algorithm on an evaluation metric. Using an algorithm without tuning its parameters would result in possible suboptimal performance of the algorithm.

GridSearchCV was used for parameter tuning of the Decision Tree classifier. The criterion parameter was tuned for either 'gini' or 'entropy' and the minimum sample split parameter was tuned in the range from 2 to 20. The best estimator from GridSearchCV used a criterion of 'entropy' and minimum sample split of 5.

Validation

The algorithm was validated using cross validation in GridSearchCV. Specifically, validation was performed using StratifiedShuffleSplit. StratifiedShuffleSplit shuffles the data and splits it while retaining the same percentage of target classes as the whole set. The number of splits was set to 100 and the test size was set to 0.1.

Performance Metrics of Final Algorithm

The decision tree classifier had an accuracy of 0.87, precision of 0.51 and recall of 0.46. The accuracy of the classifier measures the proportion of people correctly identified as persons of interest or not persons of interest out of all predictions. The precision would show the proportion of identified persons of interest who are actually persons of interest. The recall shows the proportion of actual persons of interests who are correctly identified by the classifier.