Project 5 – Machine Learning

Jason Cabral

1. Project Summary

The goal of this project was to create a model that compares a number of known financial and email-related data points for 145 Enron employees and attempt to identify if any of the employees not labeled as a person of interest in law enforcement investigations have enough in common with those who are, to potentially be labeled as such and warrant additional investigation. The data was compiled from documents surrounding various court cases after the collapse of the energy company in 2001.

The data contained 21 features, including one that identified each person as a person of interest ('poi'), of which there were 18. The feature 'loan_advances' was found to have very few data points, so it was removed. Errors in running the code lead me to find that two entries ('BHATNAGAR SANJAY' and 'BELFER ROBERT') had shifted data, which was corrected manually. One entry in the initial dictionary was the total for all features in the data set. As these were not legitimate data points to be considered for classification, the entry was removed. There were a few major outliers in the dataset, but most were kept intact as they represent accurate data who's scale may help identify person's of interest. Finally, Steven J Kean was identified as having potentially outlier-levels of incoming email messages and in fact his entry was was detrimental to the performance of the model, so it was removed.

2. Features

Of the initial features provided by the dataset, only 'loan_advances' was removed, due to it's lack of useful information. I suspected the presence of latent features that may help to identify employees as persons of interest, so I utilized Principal Component Analysis to help find these features. PCA thrives from having as much data as possible to start with, so all but 'loan_advances' were left in the data set. Additionaly, four features were created based in intuition. First, I 'normalized' total_payments to get a better sense of single year compensation by excluding such things as deffered payments, loans and expenses. Next, I used this new feature to calculate the percentage of total compensation that was allocated to stocks, rather than paid out in cash. Finally, I calculated the percentage of total incoming and outgoing emails that involved a previously identified person of interest.

Of these 4 new features, only out_poi_percentage had a positive effect on the Recall Score of a test model. With none of the new features included, the base recall score was 0.2705. Including normalized_payments and in_poi_percentage separately reduced the recall score to 0.252. Adding only stock_ratio resulted in a relatively low recall score of 0.201, but when combined with out_poi_percentage (0.2805), the test model produced the high score of 0.297.

A standardizing scaler was chosen in order to compare each feature on a level plane, rather than on a common scale. This prevents dominance by features with the greatest variance and conditions the data for PCA. After PCA transformed the data, I implemented univariate feature selection with SelectKBest. The scaler, PCA transformation and SelectKBest feature selection were implemented using a Pipeline and some parameters were optimized by GridSearchCV and manual testing. After scaling, 3 Principal Components were kept and of these, all were selected for the classifier with the corresponding scores: [4.41, 0.83, 0.06].

3. Algorithm

The algorithm used in the final model was the Linear Support Vector Classifier because I was classifying non-text, labeled data and we had less than 100,000 samples in the dataset. After initially attempting and failing to implement a basic Support Vector Classifier, I found more success in a Linear Support Vector Classifier and K Nearest Neighbors Classifier. During primary exploration, KNN produced better results for precision and accuracy, but my target was recall score and the tuned LinearSVC produced a higher recall score than the KNN model. The highest accuracy achieved with the KNN model was 0.8236, while the final accuracy for the LinearSVC model was 0.681. More importantly, the LinearSVC model bested the KNN model's recall score 0.488 to 0.365. With 15,000 predictions, this corresponds to 1024 False Negatives from the LinearSVC model, compared to 1270 from the KNN model.

4. Parameter Tuning and Selection

The scaler, feature transformation, feature selection function and classifying algorithm all have parameters that can be tuned to alter the target or method of the process. Selecting incorrect parameters or leaving the defualts can not only result in painfully slow code, but the results of the model could either be poor or not ideal for the intended target metric. In order to mazimize the performance of my model, I implemented GridSearchCV parameter testing and selection.

While I used GridSearchCV as a starting point, I found that I was able to significantly improve the target metrics with some manual tuning. The best performance I achieved from an exhaustive GridSearchCV build, with the scoring parameter set to 'recall', was a recall score of 0.28 and an accuracy score of 0.83607. By manually tuning 'max_iter' for LinearSVC, 'n_components' for PCA and 'k' for SelectKBest, I was able to achieve a recall score of 0.488 and while the accuracy score was 0.681. Other parameters for the final model that were optimized with GridSearchCV were all of the parameters for StandardScaler, along with 'penalty' and 'loss' for LinearSVC.

5. Validation

Validation in machine learning is the process of training and testing the model on different subsets of the initial data. This is extremely important as it helps prevent overfitting the model to the initial data and also helps to estimate the performance of the model on an independent set of data.

Cross validation was implemented in my model in two ways. First, the data set was split into training and testing data and the proportion of test data was set to 31%, which I found to produce the best final result. The classifier was trained on 98 individuals and the performance of the model was tested on 45 individuals. Second, in the parameter tuning stage, Kfold cross validation was implemented in the GridSearchCV function with 5 folds.

6. Performance

The final model's evaluation metrics are as follows:

Accuracy: 0.68100 Precision: 0.20604 Recall: 0.48800 F1: 0.28974 F2: 0.38314

The primary metrics used to evaluate the performance of the model are Accuracy, Precision and Recall. Accuracy is the proportion of total items in a class that are labeled or predicted correctly. Precision is the probability of a given prediction to be accurate. Recall is the percentage of a given class that is predicted accurately. As the goal of the project was to "identify Enron Employees who ***may*** have committed fraud", I was mostly concerned with the Recall Score of this model. Because we are not identifying those who are definitively guilty and simply flagging persons for further investigation, it is more important, in my opinion, to limit the number of False Negatives (clearing an actual POI) than to limit False Positives (investigating an innocent person) and this is done by targeting the highest possible recall score. As I've stated previously, this model had 1024 False Negatives and this means that of the 15,000 POI predictions in the evaluation, 1024 were not correctly labeled as a POI. In addition to Accuracy, Precision and Recall, the model achieved relatively low F1 and F2 scores along with a poor Area Under Curve score, indicating that the model was not a good fit.