Enron Fraud Detection

By Jiffin James

1-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 us 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?

Ans-The goal of this project is to identify Enron employees who may have committed fraud based on the public Enron financial and email dataset.

Dataset Background-

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. POI(Persons Of Interest)-It is 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.

Total number of data points(Number of Employees)=146

Allocation across classes(POI/non-POI)=18 (POI),128 (Non-POI)

Number of features used=21

Features={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}

I used all these features except email_address because it is unique for everyone and it won't be of much use for our machine learning algorithms.

Number of missing values for each feature

{salary=51,to_messages=60,deferral_payments=107,total_payments=21,exercised_stock_options=44,bonus=64,restricted_stock=36,shared_receipt_with_poi=60,restricted_stock_deferred=128,total_stock_value=20,expenses=51,loan_advances=142,from_messages=60,other=53,from_this_person_to_poi=60,poi=0,director_fees=129,deferred_income=97,long_term_incentive=80,email_address=35,from_poi_to_this_person=60}

Outliers

From the outliers lesson,we found that there is an outlier named "TOTAL". I removed it using pop function.It is a sum variable so it has a very huge value which can affect our results.

“THE TRAVEL AGENCY IN THE PARK” too looks like an outlier because it can never be a person’s name.

There was an entry in the dataset with all its values missing (“LOCKHART EUGENE E”).I removed both of the above entries.

2. What features did you end up using in 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 importances 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”, “intelligently select features”, “properly scale features”]

Ans-Scores of all the features-

salary	(Score=18.2896)
to_messages	(Score=1.6463)
deferral_payments	(Score=0.2246)
total_payments	(Score=8.77277)
exercised_stock_options	(Score=24.8150)
bonus	(Score=20.7922)
restricted_stock	(Score=9.2128)
shared_receipt_with_poi	(Score=8.5894)
restricted_stock_deferred	(Score=0.0654)
total_stock_value	(Score=24.1828)
expenses	(Score=6.0941)
loan_advances	(Score=7.1840)
from_messages	(Score=0.1697)
other	(Score=4.1874)
from_this_person_to_poi	(Score=2.3826)
director_fees	(Score=2.1263)
deferred_income	(Score=11.4584)

long_term_incentive	(Score=9.9221)
from_poi_to_this_person	(Score=5.2434)
ratio_from_poi_to_this_person	(Score=3.1280)
ratio_from_this_person_to_poi	(Score=16.641)

From the above scores it is quite clear that the new feature ratio_from_this_person_to_poi scored very well.In fact,it was the 5th best feature which means it was one of the important features.Thus,it was included in the final set of features.The feature ratio_from_poi_to_this_person scored very low and thus was not included.

To select the best value of k for the SelectKBest algorithm,I manually ran SelectKBest algorithm with different k values.The scores are as follows-

K	Accuracy	Precision	Recall
1	0.9040	0.4605	0.3210
2	0.8406	0.4688	0.2675
3	0.8430	0.4858	0.3510
4	0.8467	0.5031	0.3230
5	0.8562	0.4954	0.3265
6	**0.8605**	**0.5157**	**0.3855**
7	0.8542	0.4871	0.3795
8	0.8539	0.4861	0.3950
9	0.8410	0.3835	0.3170

From k=10, All values were decreasing.So I didn't include it in the above table.

Since **k=6** gave me the best possible combination of all 3 values i.e Accuracy,Precision and Recall,hence I used the following top 6 features (in terms of scores).

- salary (Score=18.575)
- bonus (Score=21.060)
- exercised_stock_options (Score=25.097)
- total_stock_value (Score=24.467)
- deferred_income (Score=11.458)

Yes, I performed feature scaling using MinMax scaler,because the features needed to be standardized .

Features Engineered

1. ratio_from_this_person_to_poi= from_this_person_to_poi / from_messages
2. ratio_from_poi_to_this_person= from_poi_to_this_person_to_poi / to_messages

Because POIs send/receive emails to/from other POIs at a rate higher than for the general population.

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

Ans- I ended up using Naïve Bayes Classifier.Other algorithms used-Decision Tree and Random Forests .

Naïve Bayes

(Using tester.py)

Accuracy=0.8605 or 86.05%

Precison=0.5157

Recall=0.3855

Decision Tree Classifier

(Using tester.py)

Accuracy=0.7927 or 78.83%

Precison=0.2784

Recall=0.2835

Random Forest

(Using tester.py)

Accuracy=0.8544 or 85.44%

Precison=0.4748

Recall=0.1790

I ended up using Naïve Bayes because it was far more consistent than the above two algorithms.Both the algorithms(Decision Tree and Random Forests) performed poorly when tester.py was executed.Random Forest was slowest among all these algorithms,this could be

explained by the fact that it creates several decision trees which is why the processor comes under huge load whenever random forest is being implemented.

4. 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? What parameters did you tune? (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 items: "discuss parameter tuning", "tune the algorithm"]

Ans-Parameter tuning is the process of finding the best combination of parameters.If parameter tuning is not performed properly,we might end up getting poor results in Accuracy,Precison and Recall.**In my case,since I used Naïve Bayes,hence I didn't perform any parameter tuning(still I have included the code for parameter tuning of decision tree)**.I would have used GridSearchCV to tune following parameters of a decision tree classifier - Criterion(gini,entropy) ,max_depth(None,1,2,3) , min_samples_split(2,3,4,5).

5.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 items: "discuss validation", "validation strategy"]

Ans-Validation is the process of checking how well the model works by splitting the data into training and testing sets.Often it happens that models work pretty well on training sets but when it comes to testing sets,the performance dips.This is also sometimes referred to as overfitting.I used train_test_split function for cross validation.The dataset was divided into two sets training set (70%) and testing set (30%). Overfitting is the classic mistake that we tend to commit whenever validation is done wrong.Eg-Not using testing set or assigning a very small part of the whole dataset as testing set (around 5-10%).

6.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"]

Ans- Precision and Recall are the most widely used evaluation metrics.The Precision for my model was 0.5157 and Recall was 0.3855.Precision is the fraction of relevant instances among the retrieved instances which in our case is 0.5157 i.e Around 51.5% of the total retrieved instances were relevant.Similarly Recall is the fraction of relevant instances that have been retrieved over the total amount of relevant instances which in our case was 0.3855 or 38.5%.