**ENRON PERSON OF INTEREST DETECTOR**

Raja Ranjith G

+91-9600105347

(rajaranjith@gmail.com)

**1.ABSTRACT**

This report describes our efforts to detect the fraudents in one of the largest companies in United States Enron

in 2000. This report provides the insights into how the data was analyzed and cleaned to apply different machine learning

algorithms. We have used Naive Bayes & Decision Tree Classifier to train and predict Enron Dataset and compared the

result from both the algrithms.We have used 3fold

cross validation and establised the why accuracy is not the correct

method to find out goodness of our result and provided the appropriate method to do so. We have also concluded why is

Naive Bayes algorithm fails to give good results on this dataset and provided the future areas in which the work can be

extended.

**2.INTRODUCTION**

**In 2000, Enron was one of the largest**

**companies in the United States. By 2002,**

**it hadcollapsed into bankruptcy due to**

**widespread corporate fraud. In the**

**resulting Federal investigation, there was**

**a significant amount of typically**

**confidential information entered into**

**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 as a result of the Enron**

**scandal. To assist you in your detective**

**work, we've combined this data with a**

**handgenerated**

**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.**

**W e have investigated the Enron e mail**

**corpus together with employee financial**

**data to identify employees' involved in**

**the fraud (hereto referred as a person of**

**interest or poi).**

**3.DATASET**

**Measurement Value**

**Total Data Points 146**

**Point of Interest**

**Data points**

**18**

**Non Point of**

**Interest Data**

**points**

**128**

**Number of Features 21**

**Handling missing values:**

**Features with more than 50% missing values:**

deferral\_payments ,

loan\_advances , restricted\_stock\_deferred ,

deferred\_income, long\_term\_incentive,

director\_fees

**POI’s with more than 50% missing values:**

deferral\_payments ,

loan\_advances , restricted\_stock\_deferred ,

director\_fees

**Conclusion:**

The features r estricted\_stock\_deferred,

director\_fees , loan\_advances

have both a large number of missing nonpoi(>

110) and

poi(>17); therefore it would be imprudent to use them as

features.

**Initial Dataset:**

Dataset after financial features were adjusted:

**4.RELATED WORK**

● A paper describing the Enron data was

presented at the 2004 CEAS conference.

● Some experiments associated with this data are

described on Ron Bekkerman 's home page.

● Structure of data in enron dataset

http://research.cs.queensu.ca/~skill/enron.pdf

**5.APPROCH**

There are four major steps in our project:

1. Enron dataset

2. Feature processing

3. Algorithm

4. Validation

First of all We'd like to have a look at my data and check

it for outliers. We plot salaries and bonuses on Enron

employees and see an outlier in the data

When We checked it We see this is a number of total

salary and bonus. As this is not sensible information for

our analyss. We removed it manually. Two more outliers

(SKILLING JEFFREJ and LAY KENNETH) We kept in

dataset as these values are real and actually they are

already a sign of these two managers being involved in

the fraud. Now dataset looks like this:

**5.1 Feature Processing**

After cleaning the data from outliers We had to pick the

most sensible features to use. First We picked

'from\_poi\_this\_person' and 'from\_this\_person\_to\_poi' but

there was no strong pattern when We plotted the data so

We used fractions for both features of 'from/to poi

messages' and 'total from/to messages'.

POI's in the dataset. There were 35 people who were

Two new features were created and tested for this

project. These were:

● the fraction of all emails to a person that were

sent from a person of interest.

● The fraction of all emails that a person sent that

were addressed to persons of interest.

Our hypothesis was that there is stronger connection

between POI's via email that between POI's and

nonPOI's.

When we look at scatterplot we can agree that

the data pattern confirms said above, eg. There is no POI

below 0.2 in 'y' axis.

In order to find the most effective features for

classification, feature selection using “Decision Tree” was

deployed to rank the features. Selection features was half

manual iterative process. First We put all the possible

features into features\_list and then started deleting them

one by one using score value and human intuition.

**We picked 10 features which are:**

[ **“salary”, “bonus”,**

**“fraction\_from\_poi\_email”,**

**“fraction\_to\_poi\_email”, deferral\_payments”,**

**“total\_payments”, “loan\_advances”,**

**“restricted\_stock\_deferred”,**

**“deferred\_income”, “total\_stock\_value”** ]

**Accuracy** for this feature set is around 0.8

Approximate **feature ranking** :

It

But with these features our precision and recall were too

low( less than 0.3) so We had to change my strategy and

manually pick features which gave me precision and

recall values over 0.3 . In this dataset We cannot use

accuracy for evaluating my algorithm because there are

few POI's and in dataset and the best evaluator are

precision and recall. There were only 18 examples of

POIs in “real life”, but for various reasons, half of those

are not present in this dataset.

Finally We picked the following features:

[ **“fraction\_from\_poi\_email”,”fraction\_to\_poi\_email”,”**

**shared\_receipt\_with\_poi”** ]

**5.2 Algorithm selection and Tuning**

Firstly We tried Naive Bayes, accuracy was lower than

with Decision Tree( 0.267 and 0.92 respectively). We

made a conclusion that the feature set We used does not

suit the distributional and interactive assumptions of

Naive Bayes well.

We selected Decision Tree Algorithm for the POI

identifier. It gave me accuracy before tuning parameter in

range 0.690.83

and after tuning to 0.890.93.

No feature scaling was deployed, as it's not necessary

when using a decision tree.

After selecting features and algorithm We manually tuned

parameters **min\_samples\_split** .

It turned out that the best value for min\_split\_value are 5

and 6.

**5.3 Analysis Validation and Performance**

This process was validated using 3fold

crossvalidation,

precision and recall scores. First We used accuracy to

evaluate my algorithm. It was a mistake because in this

case we have a class imbalance problemthe

number of

POIs is small compared to the total number of examples

in the dataset. So We had to use precision and recall for

these activities instead.

We was able to reach average value of **precision=0.667**

and **recall=0.667**

**6.RESULTS AND DISCUSSION**

The precision can be interpreted as the likelihood that a

person who is identified as a POI is actually a true POI;

the fact that this is 0.67 means that using the identifier to

flag POIs would result in 32% of the positive flags being

false alarm. Recall measures how likely it is that identifier

will flag a POI in the test set. 67% of the times it would

catch the person and 33% of the time it wouldn't.

**7.CONCLUSION AND FUTURE DIRECTION**

These numbers are quite good but we still can improve

the strategy. One of the possible paths to improvement is

digging in to the emails data more. The email features in

the started dataset were aggregated over all the

messages for a given person. By digging in the starter

dataset we aggregated over all the messages, it's

possible that more detailed patterns(say, messages

to/from a specific address, rather than just messages

to/from any POI address, or the usage of specific

vocabulary terms) might emerge. Since we live in a world

in which more POI finance data might not be easy to find,

the next realistic thing to try might be to extract more data

from the emails.

**ACKNOWLEDGEMENT**

We thank Dr. Atul Gupta for getting us started, and

helping us along the way.

We would along like to acknowledge CMU for providing

the Enron Data Set.

**REFERENCES**

● Udacity Introduction

to machine

learning (ud120)

● ScikitLearn

machine learning library

● Numpy – A scientific calculation python

library

● Matplotlib – For plotting graphs

● Enron data set

https://www.cs.cmu.edu/~./enron/