1. **Dataset and Questions**

The goal of this project is to identify persons of interest from email corpus of Enron Data Set.

The POIs are people who are suspicious and have been or may have been involved in the fraud.

We have huge data set and list of POIs who were directly or indirectly involved in the fraud.

We can take advantage of this publicly available data and use Machine Learning Techniques to

find pattern in data and then classify a person as POI or Non-POI.

***Data Exploration:***

* No of Data Points:

Total Number of Data Points available: **146**

We have data available for 146 persons.

* List of available features

List of All Features available: ['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']

* POIs/Non-POIs

Number of POIs: 18

Number of Non POIs: 128

* There are a few features that have more than 50% missing data and might be worthwhile to exclude:
* loan\_advances
* director\_fees
* restricted\_stock\_deferred
* deferral\_payments
* deferred\_income
* long\_term\_incentive

Also 'email\_address' feature is a textual feature and can be removed as it doesn’t

hold significant information.

After looking over the insiderpay.pdf file, I decided to remove ‘THE TRAVEL AGENCY IN THE

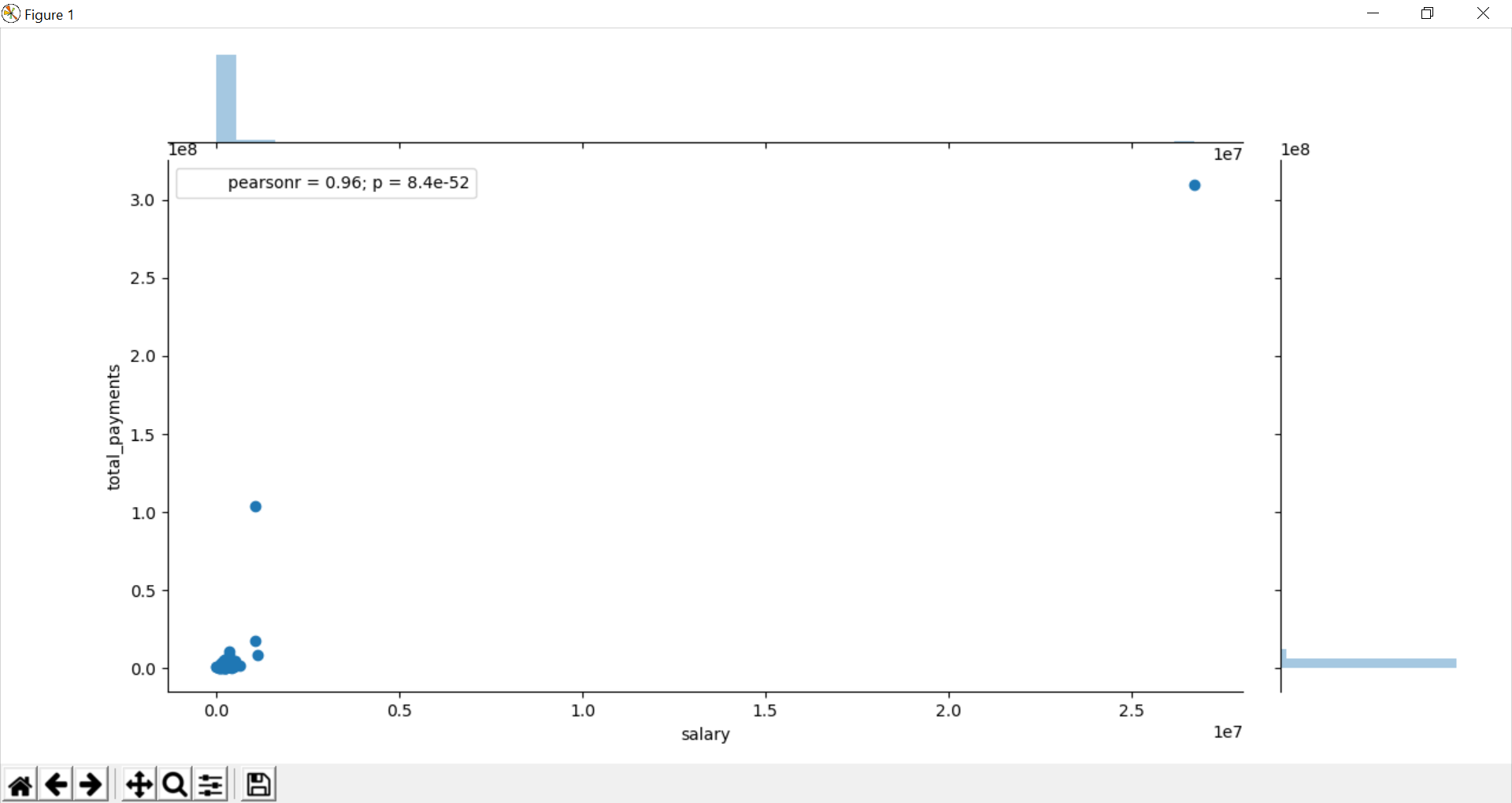
PARK’ observation as its not a person.

***Outlier Detection:***

A graph was plotted salary vs total payments for outlier detection.

[LAY KENNETH L, TOTAL] were detected as outliers. Ken Lay is a POI and is retained in the dataset.

TOTAL was removed from the data set.



1. **Feature Engineering/ Feature Selection**

* During data exploration phase some features having a lot of missing values were removed.

Also, the ‘email\_address’ feature was removed as it didn’t contain any vital info.

* Below two new features have been created:

percentage\_msg\_to\_poi, percentage\_msg\_from\_poi.

These two features indicate the percentage of emails sent to or received from poi.

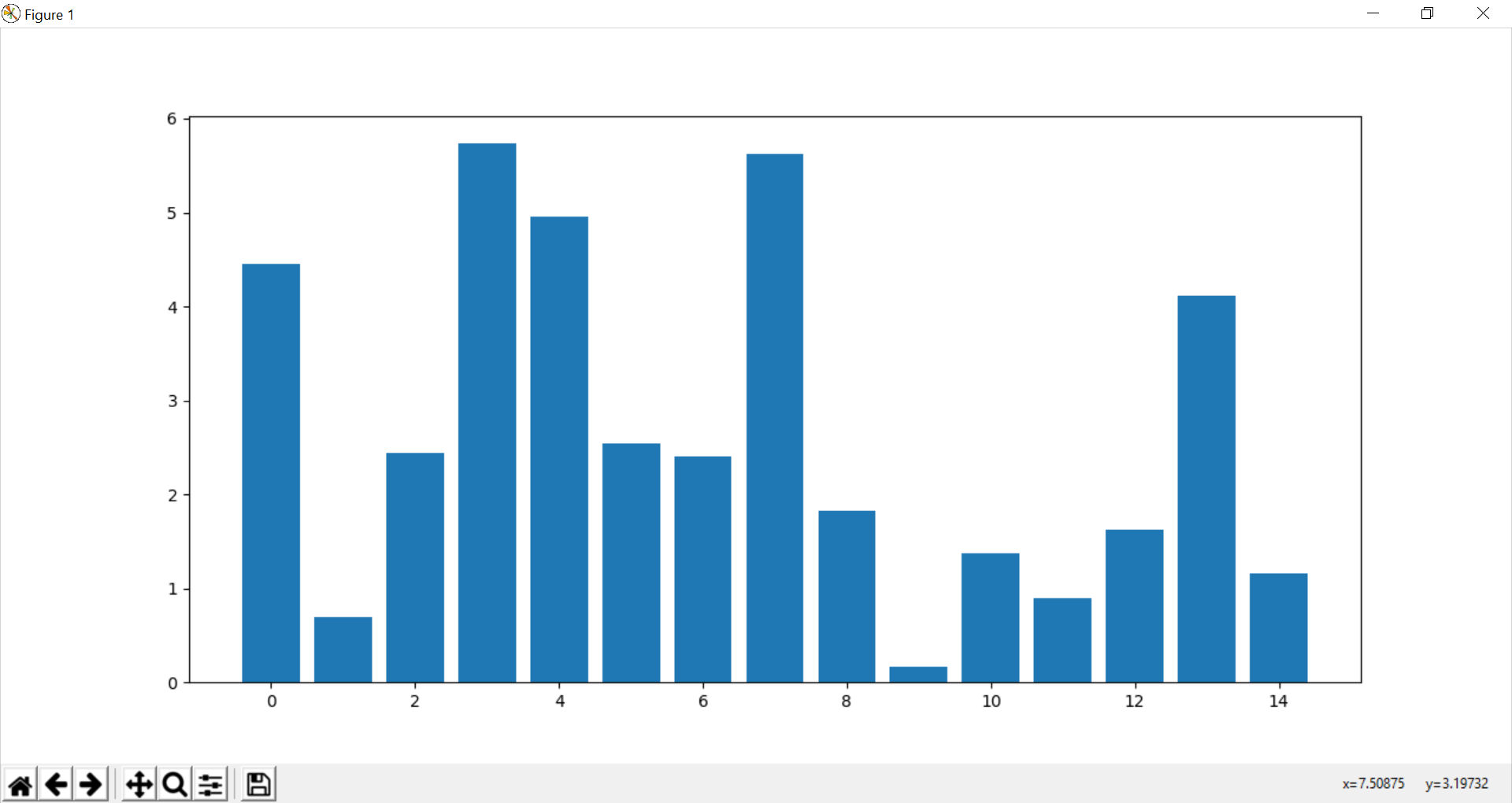
The above two features give a better insight to that data and provide vital info.

If a person has sent/received majority of mail from POI , he/she is likely to be a POI.

* Fetaure Scaling

We did not opt for feature scaling as we will be using algorithms like Decision Trees, Naïve Bayes .

* After manually selecting the features, we decided to use SelectKBest to do further feature selection. We calculated the feature scores and plotted it:



From the above plot we can assume that there are 9 best features. Also, the newly created

feature percentage\_msg\_to\_poi (index 13) has pretty good feature score.

I selected the top 9 features from the above feature scores and the below feature list was

finalized:

['salary','total\_payments','exercised\_stock\_options','bonus','restricted\_stock','shared\_receipt\_with\_poi','total\_stock\_value','expenses', 'percentage\_msg\_to\_poi']

Note: ‘poi’ is label.

1. **Pickup and Tune an alogorithm**

I decided to use below algorithms:

* Naïve Bayes
* DecisionTreeClassifier
* Adaboost

I tried above three algorithms and evaluated the performance.

After trying, the above three algorithms, it was found that accuracy didn’t vary

much (approx. 85% in all cases) but precision and recall varying significantly across the three algorithms.

AdaBoost gave highest precision(0.63) but the recall was quite low(0.14).

Naïve Bayes had Precision: 0.36425 and Recall: 0.24150.

Best precision and Recall was offered by Decision Tree Classifier; Precision: 0.35394,Recall: 0.32350 and hence decided to settle with Decision Tree Classifier.

1. **Parameter Tuning**

Parameter Tuning is an important part of model building process in machine learning.

Parameter tuning means selecting best values for different parameters for an algorithm.

Parameter tuning helps in optimizing the performance of a machine learning algorithm.

An algorithm can perform well with default parameters but the performance can be

Improved further by tuning the parameters carefully. If the parameters are not tuned

well we may not get the optimal performance from an algorithm.

I used GridSearchCV to tune the parameters. GridSearchCV is a great approach

to find the best parameters as one can provide a set of different parameters value

to GridSearchCV and it will return the best set of parameters.

1. **Validation**

Validation is the process of evaluating the performance of an algorithm on unseen data.

An algorithm is trained on training set and tested on test data.

The performance is validated against running the algorithm on test data.

I used stratified shuffle split cross validation to validate my algorithm. This method of cross-validation ensure that each test and train set have a balanced proportion of the target class. This is important as an imbalance can give unrealistic evaluation metrics. The shuffle split ensures randomization across folds and in such a way that folds are not continuous chunks of data .

1. **Evaluation Matrix**

The final Decision Tree classifier had an accuracy of 83 %. This means that 83% of the

Predictions made by the classifier will be correct. This is actually a poor measure of

performance in our case as the percentage of POI is much less than the percentage of

Non-POIs in our data set.

Precision and recall are better measures in this case. The precision and Recall for

The final algorithm is found to be 0.35 and 0.32 respectively.

Precision of 0.35 means that in 35% of the people labelled as POI will be an actual POI.

Recall of 0.32 means algorithm is able to identify 32% of the POIs. Rest 68 % escaped

being identified as POI and thus could not be recalled.