1. Data

1.1 The database description

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. In this project, we will play detective, and put our 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 us in our detective work, the authors have combined this data with 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.

1.2 Summary

The dataset consists of 146 data points with 21 features. 18 records are labeled as persons of interest.

1.3 The features in the data

financial features: ['salary', 'deferral_payments', 'total_payments', 'loan_advances', 'bonus', 'restricted_stock_deferred', 'deferred_income', 'total_stock_value', 'expenses', 'exercised_stock_options', 'other', 'long_term_incentive', 'restricted_stock', 'director_fees'] (all units are in US dollars).

email features: ['to_messages', 'email_address', 'from_poi_to_this_person', 'from_messages', 'from_this_person_to_poi', 'shared_receipt_with_poi'] (units are generally number of emails messages; notable exception is ‘email_address’, which is a text string).

POI label: [‘poi’] (boolean, represented as integer).

Definitions (financial features):

(salary) Reflects items such as base salary, executive cash allowances, and benefits payments.

(bonus) Reflects annual cash incentives paid based upon company performance. Also may include other retention payments.

(long_term_incentive) Reflects long-term incentive cash payments from various long-term incentive programs designed to tie executive compensation to long-term success as measured against key performance drivers and business objectives over a multi-year period, generally 3 to 5 years.

(deferred_income) Reflects voluntary executive deferrals of salary, annual cash incentives, and long-term cash incentives as well as cash fees deferred by non-employee directors under a deferred compensation arrangement. May also reflect deferrals under a stock option or phantom stock unit in lieu of cash arrangement.

(deferral_payments) Reflects distributions from a deferred compensation arrangement due to termination of employment or due to in-service withdrawals as per plan provisions.

(loan_advances) Reflects total amount of loan advances, excluding repayments, provided by the Debtor in return for a promise of repayment. In certain instances, the terms of the promissory notes allow for the option to repay with stock of the company.

(other) Reflects items such as payments for severance, consulting services, relocation costs, tax advances and allowances for employees on international assignment (i.e. housing allowances, cost of living allowances, payments under Enron's Tax Equalization Program, etc.). May also include payments provided with respect to employment agreements, as well as imputed income amounts for such things as use of corporate aircraft.

(expenses) Reflects reimbursements of business expenses. May include fees paid for consulting services.

(director_fees) Reflects cash payments and/or value of stock grants made in lieu of cash payments to non-employee directors.

(exercised_stock_options) Reflects amounts from exercised stock options which equal the market value in excess of the exercise price on the date the options were exercised either through cashless (same-day sale), stock swap or cash exercises. The reflected gain may differ from that realized by the insider due to fluctuations in the market price and the timing of any subsequent sale of the securities.

(restricted_stock) Reflects the gross fair market value of shares and accrued dividends (and/or phantom units and dividend equivalents) on the date of release due to lapse of vesting periods, regardless of whether deferred.

(restricted_stock_deferred) Reflects value of restricted stock voluntarily deferred prior to release under a deferred compensation arrangement.

(total_stock_value) In 1998, 1999 and 2000, Debtor and non-debtor affiliates were charged for options granted. The Black-Scholes method was used to determine the amount to be charged. Any amounts charged to Debtor and non-debtor affiliates

associated with the options exercised related to these three years have not been subtracted from the share value amounts shown.

2. Outlier Detection

This database contains indexes that hamper analysis: for example, a spreadsheet artifact 'TOTAL' or "NaN' values in the rows. The visualizing as a scatter plot can confirm it.

'TOTAL' was not being used in the analysis and thus removed at all.

The datasets with 'NaN' are incompatible with scikit-learn estimators. They assume that all values in an array are numerical. A basic strategy to use these datasets is to discard entire rows and/or columns containing missing values. But in our case this can be a reason of losing data which may be valuable (even though incomplete). A better strategy is to impute the missing values.The Imputer class provides basic strategies for imputing missing values, either using the mean, the median or the most frequent value of the row or column in which the missing values are located.

3. Algorithm selection

The highest result for the financial and mixed sets of features was shown by the KNeighborsClassifier() without imputing mean values instead of 'NaN' and without scaling the features:

1) ['poi', 'salary', 'bonus', 'exercised_stock_options', 'deferred_income', 'from_poi_to_this_person', 'from_this_person_to_poi', 'shared_receipt_with_poi'] - 83.171% accuracy, 81.806% precision, 31.250% recall;

2) ['poi', 'salary', 'bonus', 'exercised_stock_options', 'deferred_income'] - 88.293% accuracy, 78.361% precision, 30.600% recall.

The highest result for the created features was shown by the DecisionTreeClassifier(max_depth=1): ['poi', 'coefficient_bonus_salary', 'coefficient_income_total', 'coefficient_from_poi_all', 'coefficient_to_poi_all', 'exercised_stock_options'] - 89.327% accuracy, 91.476% precision, 22.000% recall.

3.1 Evaluation

Among all the sets of features and algorithms there is a really valuable find - KNeighborsClassifier() and ['poi', 'salary', 'bonus', 'exercised_stock_options', 'deferred_income', 'from_poi_to_this_person', 'from_this_person_to_poi', 'shared_receipt_with_poi'].
A combination of these options gives us a high level of accuracy (89.186%) and precision (81.806%) and a normal level of recall (31.250%)

It has a great practical meaning: with a lot of confidence we can see that it's very likely for a marked person to be a real POI and not a false alarm. This allows to speed the search process, to narrow down the suspects and to protect innocent people from suspicion.

4. Conclusion (Enron Submission Free-Response Questions)

4.1

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 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? [relevant rubric items: "data exploration", "outlier investigation"]

The goal of this project is constructing a predictive model for identifying persons of interest using many feature sets and different types of classifiers. The base for this model is the dataset with real events.

During the research incorrect values and outliers were found. Invalid values were corrected. High values associated with the generalization of data were removed, the rest outliers were remained in the database in order to avoid the loss of valuable information.

4.2

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", "properly scale features", "intelligently select feature"]

Selection of features for investigation seemed quite obvious: 1) indicators of salaries and bonuses - the basic data when considering the payments to staff, they are required to include both as they demonstrate quite a weak proportionality; 2) all

quantitative indicators of correspondence with persons of interest are required for including; 3) including 'exercised_stock_options', 'deferred_income' almost always led to an increase in all indicators of the accuracy of the predictions, so I added them in the final version too.

When thinking over the construction of features I decided that the ratio of these values can give even more information about the suspicious trends than the variables themselves.

Significant impact of introducing new variables on the algorithm DecisionTreeClassifier(max_depth=1)was confirmed by experiments. Accuracy: 0.86673 Precision: 0.50119 Recall: 0.10550 F1: 0.17431 F2: 0.12528 - the set with old features. Accuracy: 0.89327 Precision: 0.91476 Recall: 0.22000 F1: 0.35470 F2: 0.25940 - the set with new features.

Feature scaling is not a mandatory option for prediction models. The main reasons for applying it: 1) the range of values of raw data varies is really wide in our case and in several machine learning algorithms will not work properly without normalization; 2) gradient descent converges much faster with feature scaling than without it.

Standardizing and imputing values of variables tends to make the training process better behaved by improving the numerical condition. Family of algorithms that is most likely to be scale-invariant are tree-based methods. Scaling matters in the cases of k-nearest neighbors with an Euclidean distance measure, k-means, logistic regression, SVM, linear discriminant analysis, PCA, etc. We can see it in the case of KMeans(n_clusters=2) and ['poi','from_messages', 'from_this_person_to_poi', 'shared_receipt_with_poi']. Without scaling: Accuracy: 0.75711 Precision: 0.10832 Recall: 0.16400 F1: 0.13047 F2: 0.14871 With scaling: Accuracy: 0.72560 Precision: 0.15959 Recall: 0.24800 F1: 0.19421 F2: 0.22326

Scaling of cases should be approached with caution because it discards information. We can see it in the case of KNeighborsClassifier() and ['poi', 'salary', 'bonus', 'exercised_stock_options', 'deferred_income', 'from_poi_to_this_person', 'from_this_person_to_poi', 'shared_receipt_with_poi']. Without scaling: Accuracy: 0.89186 Precision: 0.81806 Recall: 0.31250 F1: 0.45224 F2: 0.35657 With scaling: Accuracy: 0.85253 Precision: 0.00467 Recall: 0.00050 F1: 0.00090 F2: 0.00061

Presented variables allowed to make a fairly accurate prediction (accuracy about 89%) without creating new variables and scaling. But the process of scaling and the introduction of new variables - the ratio between most important data parameters - have led to high performance in another area (precision).

4.3

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”]
During the work I tried to use more than 10 different algorithms, in the final version of the project I left only 8 from them to examine empirically the impact of the type of the algorithm and its parameters on the quality of predictions.

For the project, I selected only the information about the experiments with high accuracy and precision trying to improve the level of recall. The results of experiments on all parameters (precision, recall, f1, f2) varies widely.

KNeighborsClassifier() and ['poi', 'salary', 'bonus', 'exercised_stock_options', 'deferred_income', 'from_poi_to_this_person', 'from_this_person_to_poi', 'shared_receipt_with_poi'] is the final version of all experiments.

4.4

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? (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 item: “tune the algorithm”]

Models can have many parameters and the goal for the algorithm tuning is finding the best combination of these indicators.Tuning an algorithm is extremely important because different settings can have a profound effect on its performance. Experiments with the highest accuracy and influence the parameters of the algorithms shows very clearly how the classifier parameters can modify the final result.
If parameter tuning is not done well then it will result in poor accuracy and we could not get best result out of that classifier. If the model goes through too much tuning then it may happen that the model get overfit and too little tuning may result in underfitting of the model thus resulting a bad model. Parameter tuning is done in with different combination of parameters of the classifier to take the best result from the data.

4.5

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 item: "validation strategy"]

Validation is a technique for checking how our model generalizes with the remaining part of the dataset.
We have used validation technique as StratifiedShuffleSplit Cross Validation which prevented overfitting and made the model generalized. Here we put k=14000. Generally, it's a good practice to shuffle your data before splitting them into different groups, as long as the observations are independent from each other .

In the real world, target event may concentrate in a certain period or a certain geographic area. Simply splitting the data will create very large inter-group variance. i.e, some group might be event-free, some group might be full of events. Observations with different values will more likely be split into different groups after shuffling, so that the variance of model is reduced.

Shuffling is performed before splitting, so you are guaranteed the each fold will be different.The common mistake in this case is overfitting where the model performed well on training set but have substantial lower result on test set.In this project we have the great start code with very useful function for validation - test_classifier(). I did not change it at all.The parameter "folds" in this function is equal 14000 and it means the model runs 14000 times with different test sets based on the original data.

4.6

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"]

Definition:

In a classification task, a precision score of 1.0 for a class C means that every item labeled as belonging to class C does indeed belong to class C (but says nothing about the number of items from class C that were not labeled correctly) whereas a recall of 1.0 means that every item from class C was labeled as belonging to class C (but says nothing about how many other items were incorrectly also labeled as belonging to class C).In the final algorithm these metrics have the values: accuracy = 89.186%, precision = 81.806%, recall = 31.250%.

In practice it means I am confident in general in my predictions by 89.186%, the POI persons that were predicted in this model with confidence by 81.806% are really POI and the level of confidence that predicted POI person are all POI persons is 31.250%.It's my personal choice because from my point of view it's better to be confident in the precision.In the project we can see the result that many people will recognize as very perspective for researching DecisionTreeClassifier(max_depth=1): ['poi', 'coefficient_bonus_salary', 'coefficient_income_total', 'coefficient_from_poi_all', 'coefficient_to_poi_all', 'exercised_stock_options'] - 89.327% accuracy, 91.476% precision, 22.000% recall.