Udacity Project 5
Identify Fraud from Enron Email

Summary

In 2000, Enron was one of the largest companies in the United States, which collapsed in 2002 due to 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.

The goal of this project is to use financial and email data from Enron corpus to build a predictive model that could identify a “Person of Interest” (POI). Such model could be used to find additional suspects who were not inspected during the original investigation, or to find persons of interest during fraud investigations at other companies.

The dataset contains 21 features with 146 records, 18 of which are labeled as persons of interest. After visualizing the dataset, we found and removed the following outliers:
-- TOTAL
-- THE TRAVEL AGENCY IN THE PARK
-- LOCKHART EUGENE E

The following features have several NA values:

FEATURE	**NAs**
loan_advances	142
director_fees	129
restricted_stock_deferred	128
deferral_payments	107
deferred_income	97
long_term_incentive	80
bonus	64
to_messages	60
shared_receipt_with_poi	60
from_poi_to_this_person	60
from_messages	60
from_this_person_to_poi	60
other	53
salary	51
expenses	51
exercised_stock_options	44
restricted_stock	36
email_address	35
total_payments	21
total_stock_value	20

In our final dataset there are 9 features.

Features.

We composed three aggregate features:
-- fraction_from_poi (emails received from POIs)
-- fraction_to_poi (emails sent to POIs)

-- wealth (salary + total stock value +exercised stock options +bonuses)

Here we can see the impact of these new features in the classifier's performance:

GaussianNB TEST (no aggr) ::
Precision: 0.284765988416
Recall: 0.408261904762

GaussianNB TEST (+wealth) ::
Precision: 0.274799508884
Recall: 0.389833333333

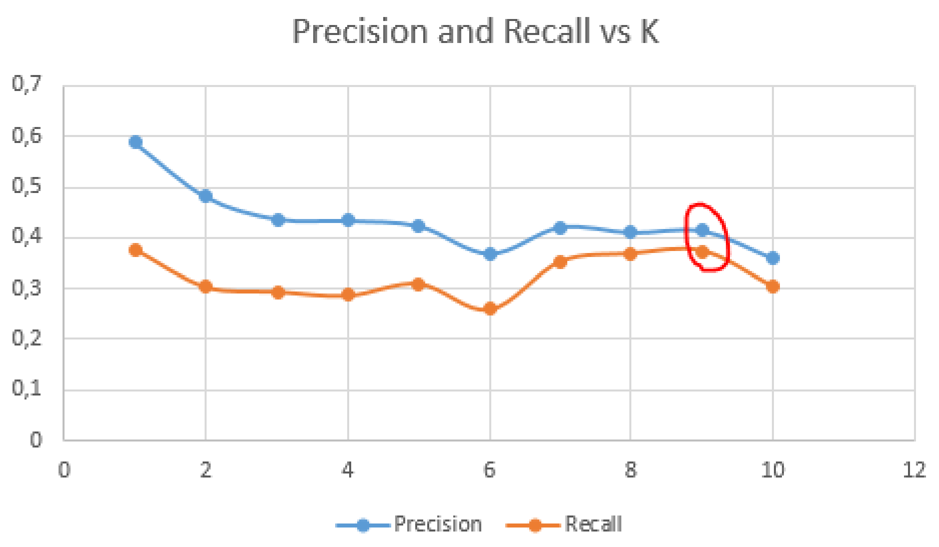
GaussianNB TEST (+fraction_from_poi) ::
Precision: 0.284765988416
Recall: 0.408261904762

GaussianNB TEST (+fraction_to_poi) ::
Precision: 0.284765988416
Recall: 0.408261904762

We also scaled all the features using the MinMaxScaler to avoid possible problems with different units in the dataset.

Finally, we used SelectKBest to select 9 most influential features:
-- 'exercised_stock_options',
-- 'total_stock_value',
-- 'bonus',
-- 'salary',
-- 'fraction_to_poi'

We can see on the plot why we decided to select exactly 9 features:



Algorithm selection

After trying 3 different algorithms we found that GaussianNB have a potential to be used further.

GaussianNB:

Precision: 0.423174603175
Recall: 0.309916666667

DecisionTree:
Precision: 0.210908730159
Recall: 0.226392857143

KNeighborsClassifier:
Precision: 0.440166666667
Recall: 0.242619047619

Algorithm tuning

Tuning means the adjustment of an algorithm when training, in order to generalize well in an independent dataset. We performed automatic parameter tuning using GridSearchCV during the selection phase.

Comparing the results with and without tuning we can see why it's so helpful:

DecisionTree (NO tuning) ::
Precision: 0.223575396825
Recall: 0.243626984127

DecisionTree (tuning) ::
Precision: 0.200706349206
Recall: 0.17453968254
Best parameters:
max_depth=2,
max_leaf_nodes=10,
min_samples_leaf=1,
min_samples_split=2,

KNeighborsClassifier (NO tuning) ::
Precision: 0.101666666667
Recall: 0.0305

KNeighborsClassifier (tuning) ::
Precision: 0.08
Recall: 0.0359285714286
Best parameters:
algorithm='ball_tree',
n_neighbors=5,
weights='uniform'

Validation

Validation is a set of techniques to check how good the model generalizes the remaining part of the dataset.

A classic mistake is over-fitting (the model performed well on training set but have much worse result on test set). In order to avoid such a mistake, we can conduct cross-validation with StratifiedSuffleSplit. The main reason of using the StratifiedSuffleSplit is the nature of our dataset, (it's extremely small with only 14 POI). A single split into a train/test set would not give a better estimate of error accuracy. Therefore, we need to randomly split the data into multiple trials.

Evaluation metrics

We used precision and recall as main evaluation metrics.

The best performance belongs to GaussianNB (precision = 0.42, recall = 0.31) which also became the final model of choice.

Precision is the ratio of records predicted as POI (true positives) to the records that are actually POI.
Recall is the ratio of true positives to records flagged as POI.

References

https://www.udacity.com/course/viewer#!/c-ud120-nd
http://scikit-learn.org/stable/documentation.html
http://machinelearningmastery.com/how-to-tune-algorithm-parameters-with-scikit-learn/
http://stats.stackexchange.com/questions/95797/how-to-split-the-dataset-for-cross-validation-learning-curve-and-final-evaluat