**Identify Fraud from Enron Email**

In 2000, Enron was one of the largest companies, which is collapsed into bankruptcy in 2002, due widespread corporate fraud, During the federal investigation the significant amount of Enron Email date made public on the web.

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?**

According to the given dataset, there are 146 persons are in the observations, *in that* ***18*** *are the* ***POI’s*** *which means ‘person of interest’ or ‘plea deal with the government’ and remaining* ***128*** *persons are the* ***Non-POI’s****.*

And each person having 21 variables, 6 are email features, 14 are financial features like *‘salary’, ’Bonus’ and so on,* and the last important one is POI.

Here we are randomly selecting some of the features and applying the machine learning concepts and algorithms to find fraud financial relations in the data set. The missing values for the selected features are:

|  |  |
| --- | --- |
| **Features** | **Missing Values** |
| poi | 0 |
| salary | 51 |
| deferral\_payments | 107 |
| total\_payments | 21 |
| loan\_advances | 142 |
| bonus | 64 |
| total\_stock\_value | 20 |
| restricted\_stock | 36 |
| restricted\_stock\_deferred | 128 |
| other | 53 |
| long\_term\_incentive | 80 |
| deferred\_income | 97 |
| exercised\_stock\_options | 44 |
| expanses | 51 |
| director\_fees | 129 |
| from\_poi\_to\_this\_person | 60 |
| from\_this\_person\_to \_poi | 60 |
| to\_messages | 60 |
| from\_messages | 60 |

*The 20 features out of 21 features all contained missing values the only feature that did not contain missing values is POI.*

In the given data, there are few outliers found like “TOTAL”, “THE TRAVEL AGENCY IN THE PARK” and “LOCKHART EUGENE E”, which doesn’t convey any information of the individual, coming to the “TOTAL’ which is the sum of the all financial variables in the data. which may affect the machine learning classifiers. So, removed the outliers from the data set.

After cleaning the data set, 144 persons are left.

1. **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. 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.**

Here, to find a relation between the POI’s and the Non-POI’s, we created two new key features they are:

1. Emails to a POI from the person/individual.
2. Emails from the POI to the person/individual.

The two features measures the ratio of from and to emails of POI’s and the individual.

Coming to the scaling, we used to select k-best by keeping different values of k=1,2,,…,20 with the help of f\_classif. And there RandomForest precision and recall values listed below.

|  |  |  |
| --- | --- | --- |
| **K-values** | **Precision** | **Recall** |
| k=1 | 0.2 | 0.15 |
| K=2 | 0.33 | 0.2 |
| K=3 | 0.7 | 0.35 |
| K=4 | 0.56 | 0.25 |
| K=5 | 0.5 | 0.25 |
| K=6 | 0.5 | 0.2 |
| K=7 | 0.43 | 0.15 |
| K=8 | 0.5 | 0.2 |
| K=9 | 0.42 | 0.25 |
| K=10 | 0.56 | 0.25 |
| K=11 | 0.5 | 0.15 |
| K=12 | 0.38 | 0.15 |
| K=13 | 0.63 | 0.25 |
| K=14 | 0.33 | 0.05 |
| K=15 | 0.6 | 0.15 |
| K=16 | 0.57 | 0.2 |
| k=17 | 0.4 | 0.1 |
| K=18 | 0.25 | 0.1 |
| K=19 | 0.6 | 0.15 |
| K=20 | 0.29 | 0.1 |

There is best rate of recall obtained for RandomForest classifier when we use best 3 features from SelectKBest, selected using cross validation. So, k=3 is used to obtain the *F****inal Features****.*

Which notified 3 best futures and their scores are:

|  |  |
| --- | --- |
| **Best Features** | **Score(*selectKbest*)** |
| Bonus | 21.060 |
| Total\_stock\_value | 24.467 |
| Exercised\_stock\_options | 25.097 |

We can observe the figure 1&2, which shows the different number of features on x-axis, accuracy, precision and recall on the y-axis.

1. **What algorithm did you end up using? What other one(s) did you try? How did model performance differ between algorithms?**

For a best performance let’s consider a RandomForest and AdaBoost classifiers, which has shown good precision rate for the selected k features from the RandomForest and low recall for the selected features for both classifiers, no matter how many features are used. Which are shown in the fig1 and fig 2, these are the tree based classifiers, So, there is no need for **Future Scaling**.

And the scores obtained for the 3 best features using the RandomForest classifier and using SelectKBest is as shown

|  |  |  |
| --- | --- | --- |
| **Best Features** | **Score(*RandomForest*)** | **SelectKBest** |
| Bonus | 0.3381 | 21.060 |
| Total\_stock\_value | 0.3344 | 24.467 |
| Exercised\_stock\_options | 0.3273 | 25.097 |

*bonus, total\_stock\_value, exercised\_stock\_optins* are **Final Features,** used for future on 144 persons, obtained importance of these features using RandomForest model as shown in above table when we fitting as best features.

1. **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?**

The important crucial part of the machine learning concepts has tuning the algorithms. It is used to find the optimal set input parameters by changing the input values in the algorithms. If it is not tuned well then, the algorithms will *over/underfit* the data which results in the suboptimal results.

In our case to get a high recall score using the hyper-parameters for a RandomForest and AdaBoost classifiers, we used GridSearchcv which helps us to find the complexity of the classifiers and finding the right values to perform better.

'max\_features': 3, 'min\_samples\_split': 2, 'n\_estimators': 100 are the values tuned for RandomForestclassifier.

'n\_estimators': 80, 'learning\_rate': 1 are the values tuned for AdaBoost classifier.

1. **What is validation, and what’s a classic mistake you can make if you do it wrong? How did you validate your analysis?**

Validation is a process of training and testing the algorithms to get a reliable performance results behalf of overfitting. And the data is split into parts like training and testing sets with the help of *StratifiedShuffleSplit* method to perform well for the unseen and the over fitting sample data.

Here stratifiedshufflessplit is used to shuffle and split the data into 1000 folds (different sets) to identify the best optimal combination of parameters, and it is also used is to counter the imbalance in the dataset.

Some of the classifier results are:

|  |  |  |
| --- | --- | --- |
|  | **RandomForest classifier** | **AdaBoost Classifier** |
| Accuracy | 0.84377 | 0.80162 |
| Precision | 0.49013 | 0.34241 |
| Recall | 0.38500 | 0.31450 |

The precision says the portion of POI’s classified and the recall explains the POI’s involved in the settlements with the government.

Precision

Recall

And as result, we can notify that RandomForest classifier is having high precision and recall than the AdaBoost classifier. So, for the result, RandomForest is chosen to be the best final classifier(it is having high rate of chances to find bankruptcy).

1. **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.**

In our project, we measured three evaluation metrics precision, recall, and F-1 score. In these case metrics are calculated by comparing the POI’s and non-POI’s with the actual ones.

The precision is calculated by dividing the positively identified POI from data (True Positive) to the total identified POI’s (True and False Positives) gives a high precision which indicates the algorithm is correct, if not NON-POI’s are flagged to be POI’s for low precision values.

A recall is measured by dividing the positively identified POI to the sum of negatively identified POI which is false (False Negative) and positively identified POI. A high value of recall explains there are POI’s which are identified with the help of algorithm, if not they missed and not identified.

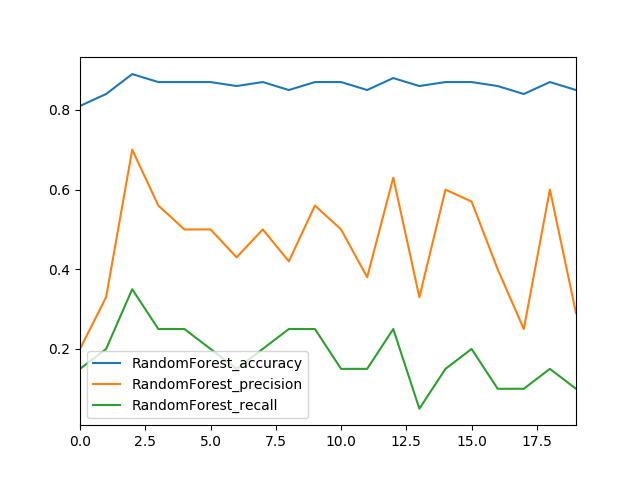
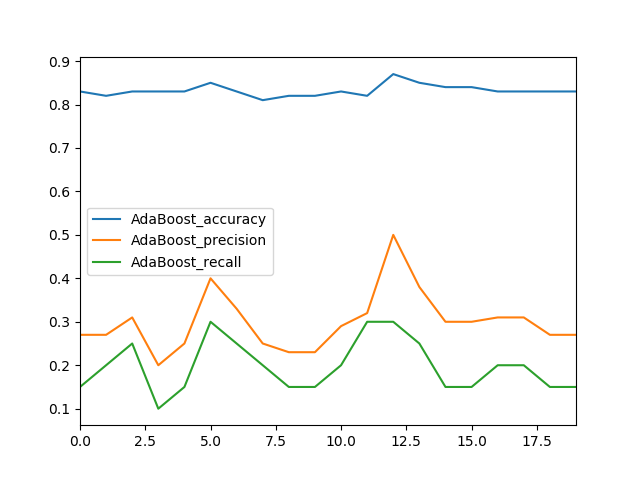
The F1 score is a weighted average of recall and precision, which obtained greater than 0.3 from the algorithms which is 0.43125 from RandomForest and 0.31450 from AdaBoost classifiers when the new features are added.

If you compare the recall and precision before adding the new features they have given us less than 0.3 from the algorithm from the RandomForest which are 0.2584(precision) and 0.1985(recall) and AdaBoost which are 0.6791(precision) and 0.2159(recall) classifiers, which shows a false assumption of the project.

|  |  |  |
| --- | --- | --- |
|  | **Without new features** | **With new features** |
| Accuracy | 0.6428 | 0.8016 |
| Precision | 0.2584 | 0.5601 |
| Recall | 0.1985 | 0.3935 |
| F1 score | 0.2899 | 0.4622 |

The new features are showing a positive result without any data leakages.

The test\_classifier in the file is used to find the performance of the algorithms.

In the last, there is confident that classifiers help in the real life to find the fraud. 

Number of Features Number of Features

Fig.1 RandomForest Fig.2 AdaBoost

Files:

README.md : project document

Final\_doc : this file project document

Figure\_1, figure\_2 : classifiers output

poi\_id.py : project code

tester.py : project evaluator

Final\_refferences.txt : references for the project.

And there are some .pkl files of the project.