Identify Fraud From Enron Email

UDACITY DATA ANALYST NANODEGREE P5

ARİF HİKMET ONAT BALTA

INTRODUCTION

Enron Corporation was one of the world's major energy, commodities, and services company based in Houston, Texas. [1] By the end of 2001 it had collapsed into bankruptcy due to widespread corporate fraud.

During its investigation, Federal Energy Regulatory Commission made about a half million emails and detailed financial data public and posted these data to the web. [2]

The main purpose of this project is to identify the persons of interest in the fraud case by using machine learning techniques. For this purpose an analytical model was build using Python and machine learning tasks are done using sklearn modules to predict whether a person could be considered as a person of interest or not.

DATA EXPLORATION

In the given Enron dataset, there are 146 people, only 18 of them are persons of interest (which makes things harder) where the rest 128 of them are innocent.

There are 21 features in the dataset, where one feature is a label and indicates whether a person is person of interest or not and all the other features are either financial data like salary, expenses, total stock value etc. or features extracted from emails such as the total number of emails a person received and sent, number of emails a person sent and received from a person of interest and so on.

OUTLIER INVESTIGATION

Through exploring the data analytically and by visually inspecting the features, I identified 3 outliers:

- TOTAL, which represents the column totals of the financial dataset and might be mistakenly added,
- THE TRAVEL AGENCY IN THE PARK, which is not a person and a spreadsheet quirk,
- LOCKHART EUGENE E., doesn't contain any feature value, full of missing values

For better predictions, these 3 outliers were removed from the dataset.

Although there were some data points which have many missing values, I decided to keep these ones due to the limited dataset. In our case, even a single feature value can be beneficial.

ADDING NEW FEATURES

As my investigation continued through the features, I realized some features might be more significant if they were used together. For example, instead of using total number of emails received and total number of email from persons of interest (poi) separately, a ratio of these two will clarify more things. So I created two new features and removed the ones used in the ratio.

New features:

$$fraction\ from\ poi = \frac{\sum emails\ received\ from\ poi's}{\sum emails\ received}$$

$$fraction\ to\ poi = \frac{\sum emails\ sent\ to\ poi's}{\sum emails\ sent}$$

SCALING

As said before, the given dataset consists of many features, some are measured in millions like total stock values, and some can be only one or two. To compare all the features in an equitable way, feature scaling is needed. Moreover, scaling will affect the following steps, for example feature selection and even impress the final predictor. Therefore min max scaler is applied to the full data set.

FEATURE SELECTION

This process is done automatically. Algorithm chose the best feature selection method, either Select K Best, Principal Component Analysis or the case where both used. It is achieved by using Pipeline and FeatureUnion functions together to feed the GridSearchCV. As a result, these procedures worked together to tune some list of parameters and found the best feature selection method and it's parameters that gives the highest score, which is Select K Best method with k=5.

Selected features and their scores are:

	Selected Features				
	Exercised Stock Options	*Bonus*	*Total Stock Value*	*Salary*	*Fraction To Poi*
Score	*24.815*	*20.792*	*24.183*	*18.290*	*16.410*

ALGORITHM SELECTION

After trying to tune some machine learning algorithms to my data by using GridSearchCV, results are as shown below:

	Algorithms				
	Gaussian Naïve Bayes	*Logistic Regression*	*Random Forest*	*K Nearest Neighbor*	*Adaboost*
f1 score	*0.385*	*0.292*	*0.281*	*0.260*	*0.313*

This results can be increased by tuning more parameters or choosing different algorithms, but for the purpose of this project, I chose Gaussian Naïve Bayes algorithm because it gives the best results in my trials.

PARAMETER TUNING

At the core, parameter tuning is the process which tries to optimize the parameters that maximizes results. GridSearchCV is the method used for parameter tuning in sklearn and basically it tries to loop through the given parameters to find the best resulting parameter.

This step is the most important step in a machine learning application. If done incorrectly, results can yield to overfitting, where a good parameter tuning can result in boosted scores. Our main aim in any machine learning application is to find the optimal parameters that are giving the best results with minimum features and time. Therefore it is very essential.

Gaussian Naïve Bayes algorithm doesn't have any parameters to tune but the other algorithms used in this project have many parameters to tune. For an example, I tried tuning "max_features" parameter of Random Forest Classifier and many more using GridSearchCV method in "poi_id.py" file.

VALIDATION

Validation is basically splitting a dataset into 2 parts as training and testing sets, giving an estimate of performance on an independent dataset. It is an important step in machine learning because it tries to prevent overfitting.

Because of the limited dataset and low number of poi labels, "Stratified Shuffle Split" method, having 100 and 1000 folds, is used in this project. The main purpose of this is to maximize testing and training scores by randomly splitting our dataset into 2 sets many times (in our case I did this process 100 and 1000 times).

EVALUATION METRICS

In the selected model (scale each feature -> select 5 best features -> implement Gaussian Naïve Bayes) we have an accuracy score of 85%, which means this model predicted the 85% of poi's. However, this metric is not ideal for our skewed dataset where we had only a few persons of interest. Therefore other evaluation metrics such as precision and recall are used in this project for evaluation.

- Selected model have a precision score of 0.43, which means whenever a poi gets flagged in the dataset, it's likely to be a real poi with a 43% probability.
- The recall score of the model is 0.35, which means when a poi shows up in the dataset, model is able to identify it 35% of the time.