ML Project Free Response Questions

By Dominic Nguyen

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 Enron scandal, publicized in 2001, was a corporate fraud involving several Enron employees. After an extensive federal investigation, numerous persons of interests (POIs) have been identified. These POIs represent those people that have been indicted, settled, or testified in exchange for immunity. In recent years, machine learning has become a very useful tool to make predictions on existing data sets. In this project, machine learning is used to classify POIs using a dataset of Enron employees' financial and email information. Several features are extracted from the information as inputs to a number of machine learning algorithms including Naive Bayes, decision trees, and k-means clustering. The classifiers' parameters are then tuned to provide optimal performance. Since POIs have already been identified through the federal investigation, the effectiveness of the chosen machine learning algorithm was able to be validated with test and training data and evaluated with the precision and recall metrics.

The dataset contains 145 employees with 18 POIs identified from the federal investigation. From an initial analysis of the data, an outlier and several missing data points were identified. The outlier was found when viewing a scatter plot of the Salary vs Bonus features. This outlier data point was labelled “TOTAL” and was a dictionary containing the sum values of each feature. The outlier was removed from the data set dictionary using the ‘pop’ method on the key value “TOTAL.” Missing data values were found by iterating through each data point for each feature and checking for ‘NaN’ values. These’ NaN’ values are then replaced with a zero. A summary of the number of missing data points can be seen in the output after running the attached po_id.py file.

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

All of the provided features, plus a created feature, except for the "email" feature was used in the initial analysis. This set of features will be referred to as Feature List A. No scaling of any of the features was required as the only classifiers used, decision tree and Naïve Bayes, are not affected by scaling. A new feature, "fraction_exercised_stock" was created by dividing the "exercised_stock_options" feature by the "total_stock_value" feature. It was hypothesized that the fraction of stocks that were exercised by an employee could play a role into whether that person was labelled a POI. Unfortunately, after analyzing the benefit of "fraction_exercised_stock", it was found that the feature importance attribute from the decision classifier and the feature score reported from SelectKBest were both 0 for the created feature. Thus, "fraction_exercised_stock" provided no benefit to the classifiers used.

Univariate feature selection with SelectKBest was employed. After using SelectKBest on Feature List A, a K value of 5 for the Naïve Bayes algorithm generated the highest F1 score. These 5 features, Features List B, had feature scores higher than 10 and were included in the final analysis for both classifiers. The cutoff point used to choose the top 5 features was from the K value given by SelectKBest. Features List B includes "deferred_income", "exercised_stock_options", "long_term_incentive", "salary", and "total_stock_value." The feature scores from SelectKBest on Feature List B and feature importances from the decision tree classifier using GridSearch on Feature List B are reported in TABLE 1.

TABLE 1

Feature	***Feature Score***	***Feature Importance***
deferred_income	*10.9*	*0.106*
exercised_stock_options	*23.7*	*0.404*
long_term_incentive	*9.3*	*0.075*
salary	*17.1*	*0.329*
total_stock_value	*23.0*	*0.085*

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”]

Naïve Bayes, nb_clf2b, was used as the classification algorithm as it resulted in the largest precision, recall, and thus F1 score. The decision tree algorithm was also tested but resulted in lower precision and recall. The performance between Naïve Bayes algorithm and decision tree algorithm on Feature List A and B can be seen in the TABLE 2 where GS = GridSearch.

TABLE 2

Feature List	Algorithm	Accuracy	Precision	Recall	F1 Score
A (20 features)	Naïve Bayes w/o GS	0.729	0.233	0.449	0.307
	Naïve Bayes w/ GS	0.850	0.412	0.303	0.349
	Decision Tree w/o GS	0.792	0.239	0.254	0.246
	Decision Tree w/ GS	0.830	0.311	0.226	0.262
B (5 features)	Naïve Bayes w/o GS	0.856	0.493	0.398	0.441
	Naïve Bayes w/ GS	0.856	0.493	0.398	0.441
	Decision Tree w/o GS	0.817	0.332	0.340	0.336
	Decision Tree w/ GS	0.817	0.362	0.366	0.365

NOTE: Naïve Bayes w/o GS (Feature List B) = Naïve Bayes w/ GS (Feature List B) because the k parameter selected by SelectKBest is the same as the default without GridSearch.

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? What parameters did you tune? (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 items: “discuss parameter tuning”, “tune the algorithm”]

Tuning or hyperparameter optimization is the process of selecting the optimal values for parameters that result in the best learning algorithm performance commonly estimated through cross-validation. If these parameters are not tuned correctly, a poor performance can result. Some algorithms such as decision trees have parameters that affect how the algorithm performs classification on the training data. Changing these parameters can result in different classification of the data which results in a different algorithm performance in terms of precision and recall. The effect of tuning can be seen in TABLE 2.

For the decision tree algorithm, GridSearchCV was used to determine the best values for the criterion and min_samples_split parameters. Though not a parameter of a specific algorithm, the K parameter of SelectKBest was also be tuned using GridSearchCV for both classifiers.

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 items: "discuss validation", "validation strategy"]

Validation is the strategy that is used to ensure machine learning algorithm generalizes data well. A classic mistake in validation would be to evaluate the algorithm on the same data that was used to classify it (testing on a training data set). This would result in overfitting which is characterized by a deceptively high algorithm performance on the training data set but a low algorithm performance on the testing dataset. Overfitting occurs when the algorithm is subjected to new data it hasn't seen before for the first time during evaluation. If validation is used on a subset of the training data, validation data, then the algorithm has already been primed with new data before evaluation.

In this project, a Stratified Shuffle Split (SSS) cross validation is used to split 70% of the data into training data and 30% into testing data. SSS is inputted into GridSearch to determine the best estimator for both Naïve Bayes and decision tree algorithms. Due to the imbalance between POI and non-POIs in the data (only 18 of the 145 people are labelled as POIs) and the small data set, SSS was selected to perform multiple test/train splits to result in a more optimal model as compared to a singular test/train split.

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"]

Accuracy, precision, and recall are calculated as evaluation metrics in the provided tester.py file. The F1 score, a weighted average of precision and recall, was also provided. These results can be seen in TABLE 2.

From TABLE 2, it can be seen that the Naïve Bayes algorithm, nb_clf2b, resulted in the highest accuracy, (0.856), precision (0.493), recall (0.398), and thus F1 score (0.441). Any time the final Naïve Bayes algorithm flags an employee as a POI, 49.3% of the time, the employee is actually a POI. In terms of recall, the algorithm will correctly identify 39.8% of all POIs (39.8% of POIs will be labelled as POIs). Lastly in terms of accuracy, the algorithm labelled POIs and non-POIs accurately 85.6% of the time.

Sources

Jayant. (2016, February) Re: Final project classic mistake made on validation [Udacity]. Retrieved from https://discussions.udacity.com/t/using-sklearn-pipeline-in-final-project/199064/8

Phanny. (2017, March 23) Re: Why does sklearn.grid_search.GridSearchCV return random results on every execution? [Cross Validated] Retrieved from https://stats.stackexchange.com/questions/269300/why-does-sklearn-grid-search-gridsearchcv-return-random-results-on-every-executi

Lejlot. (2013, October 1) Re: Build Dictionary in Python Loop - List and Dictionary Comprehensions [Stack Overflow] Retrieved from https://stackoverflow.com/questions/19121722/build-dictionary-in-python-loop-list-and-dictionary-comprehensions

Paisanco. (2015, September 14) Re: Scikit: how to check if an object is a RandomizedSearchCV or a RandomForestClassifier? [Stack Overflow] Retrieved from https://stackoverflow.com/questions/32569082/scikit-how-to-check-if-an-object-is-a-randomizedsearchcv-or-a-randomforestclass

I hereby confirm that this submission is my work. I have cited above the origins of any parts of the submission that were taken from Websites, books, forums, blog posts, github repositories, etc.