Project 7:

Question1:

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”]

Answer

We are given data from the Enron case we can use the machine learning to find the employees that did fraud. We Have use at least two of the Classifier to find the persons of interest(POI). Some background of the data we are given we are given we given list of employees with feature like salary, bonus, the number from message and more. We are also given the employee is a Person of Interest this help to separate from the Non-Person of Interest. By telling which person is POI I can get each person feature and pass them to the machine learning classifier. Before I send the data to the classifier I find out that there are 146 in the dataset, 21 features and 18 POI. After looking at the data I went and remove the outliers that are found. I found an outlier that have a salary over 25,000,000. After locate the person I found out that it was not a person but the total of all employees. I remove the outlier and try to locate more but the remaining outliers are import people from the Enron case. With that the data is ready.

Question2:

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”, “intelligently select features”, “properly scale features”]

Answer

In my first feature list I have most of the feature that are given so I can have each one with their score. I would run Kbest to see the score for each feature and compare to one another. After finding out the score I remove the low score feature and then rerun kbest to make sure that the score didn't change for the features. I did not do scaling because the algorithm did not call for scaled feature like svm. In my feature creation I create a feature that get the fraction of "from_poi_to_this_person" with "to_messages" and "from_this_person_to_poi" with "from_messages". I get 'from_poi_to_this_person' divide by the 'to_message'(that contain total send out) give you a percentage of mail send to POI which will be store in fraction_from_poi. We do the same thing 'from_this_person_to_poi' divide by 'from_message'(contain total receive message) give us a percentage of mail receive from POI this will be place into 'fraction_to_poi'. With both of these we can tell the percentage of the total mail was from or to a POI. We add both fraction_to_poi and fraction_from_poi to the feature_list. We later run selectKbest and find out that fraction_from_poi score 2.96 and fraction_to_poi score 15.94 .

‘fraction_from_poi and fraction_to_poi affect the precision and recall in a negative way. If you look below you see a table with recall and precision before the new feature and after new feature are add. You can see that in AdaBoost and Decision Tree recall and precision drop in Naïve Bayes it stay about the same. I keep the new feature in my list but later using selectKBest I remove 'fraction_from_poi' because it score low. Like I said above I use the feature _list(holding my feature) and ran them through SelectKBest to find the score for each feature then removing those feature that have low score and then rerunning selectKBest. The reason I choose the 10 features if I have less than 10 than my data will be underfit because too few feature and the reason I don't have more feature 10 feature my data will be overfitting because it haves to many features.

Before New Feature:

poi','salary','bonus','expenses','exercised_stock_options','total_stock_value',"from_poi_to_this_person",
"from_this_person_to_poi","from_messages","to_messages','loan_advances','long_term_incentive','deferral_payments','deferred_income'

	Recall	Precision
Naïve Bayes	.260	.243
AdaBoost	.31	.43
Decision Tree	.32	.31

After New Feature:

'poi','salary','bonus','expenses','exercised_stock_options','total_stock_value',"from_poi_to_this_person",
"from_this_person_to_poi","from_messages","to_messages','loan_advances','long_term_incentive','deferral_payments','deferred_income','fraction_from_poi","fraction_to_poi"

	Recall	Precision
Naïve Bayes	.261	.24393
AdaBoost	.311	.400
Decision Tree	.260	.250

Choosing Feature using KBest:

K = 15

'poi','salary','bonus','expenses','exercised_stock_options','total_stock_value',"from_poi_to_this_person",
"from_this_person_to_poi","from_messages","to_messages','loan_advances','long_term_incentive','deferral_payments','deferred_income','fraction_from_poi","fraction_to_poi"

K=15	Recall	Precision
Naïve Bayes	.261	.24393
AdaBoost	.311	.400
Decision Tree	.260	.250

K= 12

'poi','salary','bonus','expenses','exercised_stock_options','total_stock_value',"from_poi_to_this_person",
"to_messages",'loan_advances','long_term_incentive','deferred_income','fraction_from_poi',"fraction_to_poi"

K=12	Recall	Precision
Naïve Bayes	.283	.282
AdaBoost	.355	.446
Decision Tree	.28250	.263

K=10

'poi','salary','bonus','expenses','exercised_stock_options','total_stock_value',"from_poi_to_this_person",'loan_advances','long_term_incentive','deferred_income',"fraction_to_poi"

K=10	Recall	Precision
Naïve Bayes	.314	.309
AdaBoost	.39	.49
Decision Tree	.29	.31

K=8

'poi','salary','bonus','exercised_stock_options','total_stock_value','loan_advances','long_term_incentive', 'deferred_income',"fraction_to_poi"

K=8	Recall	Precision
Naïve Bayes	.31	.30
AdaBoost	.28	.35
Decision Tree	.26	.28

K=6

'poi','salary','bonus','exercised_stock_options','total_stock_value','fraction_to_poi','deferred_income'

K=6	Recall	Precision
Naïve Bayes	.515	.3855
AdaBoost	.276	.353
Decision Tree	.275	.273

After using Kbest and testing the feature we find out that K=10 have the best recall and precision out of all test. The reason I say that Naïve Bayes and Adaboost have a recall and Precision above .3 but

Decision tree recall is .29 and precision is .3 . if I keep remove more feature (example k=8 and k=6) Decision Tree recall and precision decrease so it best to use k=10 since it closer to .3.

Question3:

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"]

Answer

Naive Bayes was my first classifier that I used with the features. I also used AdaBoost and Decision Tree. AdaBoost have the best precision and recall out of the rest of the classifier. Naïve Bayes have the second best precision and recall but it just made it above .3 for both recall and precision. Decision Tree is the worst out of all three classifiers because sometime the recall is .29 or .3 but have a precision of .3

Question4:

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"]

Answer

Tuning are parameterized and modification of the parameters can influence the outcome of the learning process. This is not done will you will get overfitting your data. The goal of Tuning is to improve the prediction and fix any overfitting. In The decision tree the recall and precision was below .3 so to make reach that score we have to tune classifier. I tune the **min_samples_split** so that 13 sample are require before it split. **class_weight** make the dictionary balance. I tune the **min_samples_leaf** to 5 sample in a leaf of a node. Also I tune **max_leaf_nodes** to 10 this will make the max leaf node to be 10 . After running the code I find out the recall is .673 and precision of .354 which did better than the original recall of .306 and precision of.29. I want to see how the recall and precision change when I change the value of min_sample_split,min_samples_leaf, max_leaf_nodes. So I change the value of min_sample_split to 5, min_sample_leaf to 5 and max_leaf_nodes to 5 and recall increase by .38 more and the precision by .1. I want to change the value one more time to see if I can get a better recall and precison. So tune min_sample_split to 13, min_sample_leaf to 8 and max_leaf_node to 5 and get a recall of .756 and precision of .364 making it .045 better in recall than 2nd tune and .009 better in precision than 2nd tune. So Final tune have the best Recall at .756 and Precision .364

	Min_sample_split	Class_weight	Min_sample_leaf	Max_leaf_nodes	Recall	Precision
Original	N/A	N/A	N/A	N/A	.306	.29
1st Tune	13	"Balance"	5	10	.673	.354
2nd Tune	5	"Balance"	5	5	.711	.355
Final Tune	13	Balance	8	5	.756	.364

Question5:

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"]

Answer

Validation is a technique for assessing how the results of a statistical analysis will generalize to an independent data set. Using the same data to train and test is the classic mistake. Since the data is small and imbalance we have use StratifiedShuffleSplit. If we use the same train and test data, we lose the train data because it going to test so it best to split the data. How StratifiedShuffleSplit work by slitting the data into train and test set which return stratified randomized folds. The folds are made by the percentage of the sample of each class.

Question6:

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"]

Answer

Precision is the True positive divide by the sum of true positive and false positive. Precision Score is the relevant instances among the retrieved instances. **Precision is the identify as POI**. Recall is the true positive divide by the sum true positive plus false negative. Recall Score is the fraction of relevant instance that have been retrieved over the total amount of relevant instances. **Recall is the probability of being a POI out of all employees**. In Naive Bayes I receive a precision of .51 and recall of .38. With Adaboost I receive a precision .35 and recall of .37 . Lastly decision tree I receive a precision of .26 and a recall .26 without tuning. Decision Tree after tuning was a precision of .38 and recall of .71.

Summary

Overall this project was to give us a better understanding of machine learning by doing it hands on with the famous enron dataset. This help understands how to use KBest by creating a feature list and running them with Kbest and then removing the feature that have low score. Then you can use that new feature list to run them with the classifiers and compare how each classifiers results. Lastly tune your classifier and split the data to see how the result compare after the adjustment. This project gives the use the understanding and the important of machine learning and the enron case at the same time.