IE517 MLF F20

Module 6 Homework (Cross validation)

Yaxuan Wang (yaxuanw3)

Out[15]: The raw code for this IPython notebook is by default hidden for easier reading. To toggle on/off the raw code, click here.

Out[3]:

	ID	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	PAY_0	PAY_2	PAY_3	PAY_4	...	BILL
0	1	20000	2	2	1	24	2	2	-1	-1	...	
1	2	120000	2	2	2	26	-1	2	0	0	...	
2	3	90000	2	2	2	34	0	0	0	0	...	
3	4	50000	2	2	1	37	0	0	0	0	...	
4	5	50000	1	2	1	57	-1	0	-1	0	...	

5 rows × 25 columns

Out[4]:

	ID	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	
count	30000.000000	30000.000000	30000.000000	30000.000000	30000.000000	30000.000000	3
mean	15000.500000	167484.322667	1.603733	1.853133	1.551867	35.485500	
std	8660.398374	129747.661567	0.489129	0.790349	0.521970	9.217904	
min	1.000000	10000.000000	1.000000	0.000000	0.000000	21.000000	
25%	7500.750000	50000.000000	1.000000	1.000000	1.000000	28.000000	
50%	15000.500000	140000.000000	2.000000	2.000000	2.000000	34.000000	
75%	22500.250000	240000.000000	2.000000	2.000000	2.000000	41.000000	
max	30000.000000	1000000.000000	2.000000	6.000000	3.000000	79.000000	

8 rows × 25 columns

First, I split the data into 10% testing set and the rest 90% into training set. Then, I fit the decision tree classifier with gini as my criterion to our training set, and compute the accuracy score on the testing set by using the fitted model.

```
The accuracy score for decision tree model fitted on the testing dataset is:
0.8216666666666667
```

Part 1: Random Test Train Splits

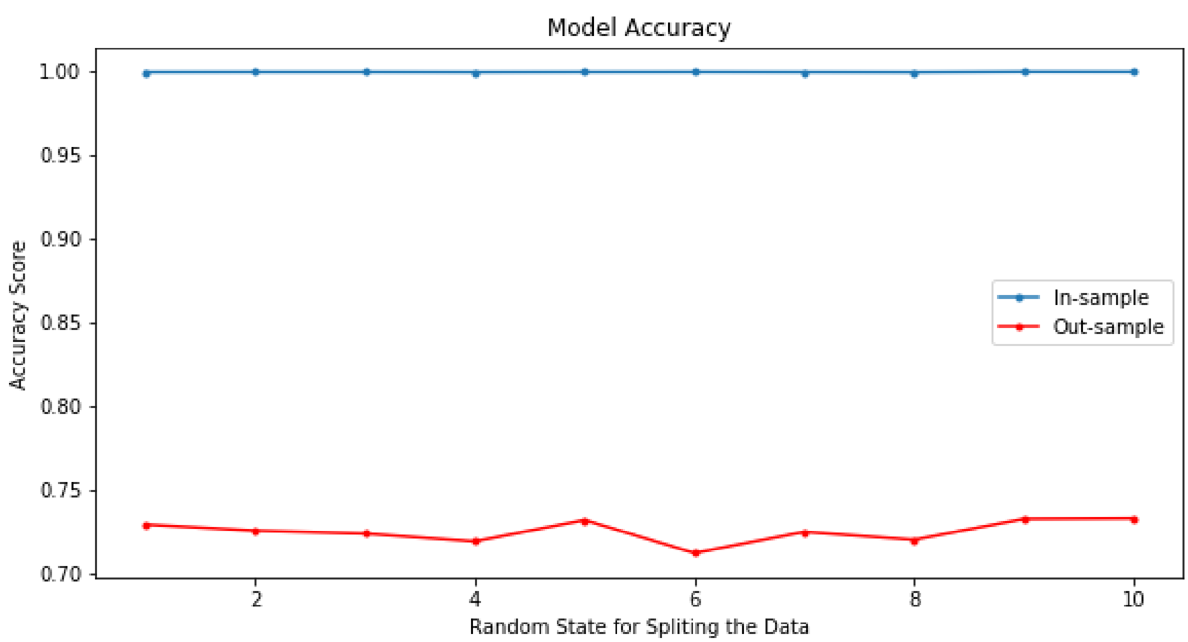
In this section, I will use the Pipeline to first impute the mission values as the average value of each feature, then normalize our data, eventually fit the decision tree classifier on testing dataset. After that, I will compute the in-sample and out-of-sample accuracy scores for 10 different samples by changing random_state from 1 to 10 in sequence.

The below tables shows the individual scores for each random state, and the mean as well as standard deviation on the set of scores.

Out[8]:

	The accuracy score changed by random_state											
	1	2	3	4	5	6	7	8	9	10	Mean	Std
In-samples	0.9993	0.9994	0.9994	0.9993	0.9994	0.9994	0.9993	0.9993	0.9995	0.9995	0.9994	0.0001
Out-samples	0.7220	0.7253	0.7293	0.7070	0.7270	0.7120	0.7247	0.7227	0.7280	0.7287	0.7227	0.0071

The below plot clearly shows the variation of accuracy scored changed by random state.



Part 2: Cross Validation

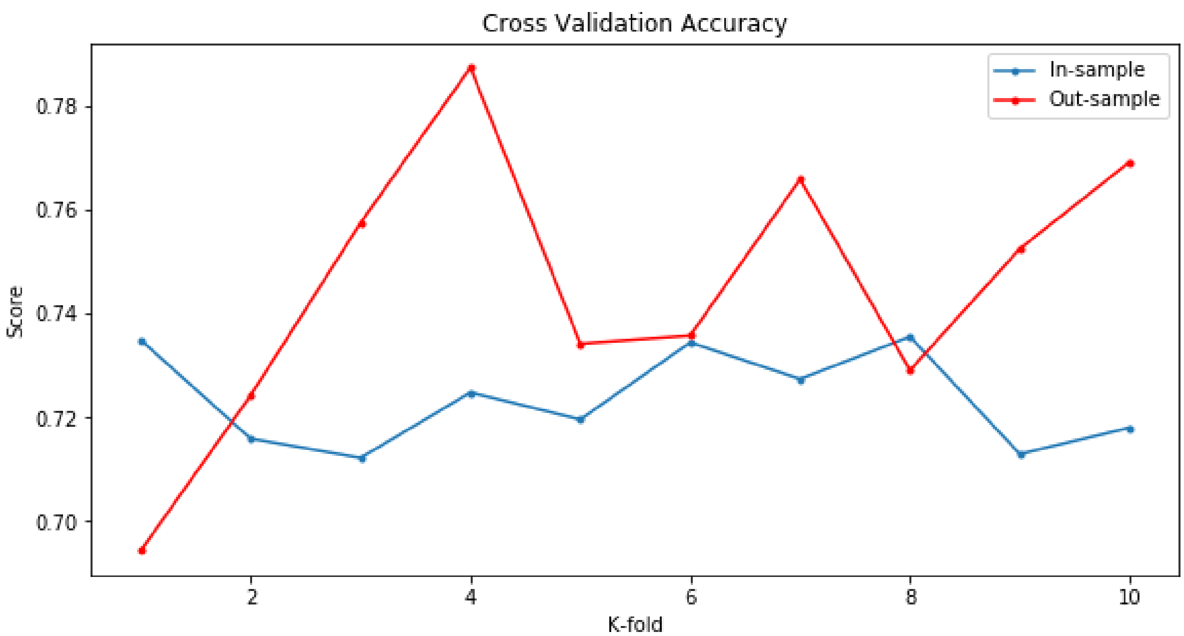
In this section, I use cross_val_scores with k-fold CV (k=10) to fit the decision tree classifier.

The below tables shows the individual fold accuracy scores, and the mean as well as standard deviation of thefold scores.

Out[12]:

The CV score changed by K-fold												
	1	2	3	4	5	6	7	8	9	10	Mean	Std
In-samples	0.7308	0.7274	0.7130	0.7319	0.7274	0.7341	0.7241	0.7396	0.7093	0.7221	0.7260	0.0088
Out-samples	0.7110	0.6977	0.7542	0.7741	0.7375	0.7124	0.7860	0.7291	0.7291	0.7458	0.7377	0.0266

The below plot clearly shows the variation of CV accuracy scored changed by number of fold.



Part 3: Conclusions

Based on the accuracy score, we can see that in the K-fold cross validation method, there are cases that the CV accuracy score for the out-sample/testing dataset are higher than the one for in-sample/training dataset. However, in the random test train splits method, there are no cases that the accuracy score for out-sample/testing dataset are higher than the one for in-sample/training dataset, which might be an overfititng issue. To best estimate on the unseen data, I suggest to use K-fold cross validation everytime before fitting the model.

The method of random test train splits is more efficient to run. This is because k-fold cross validation will split the data 10 times to achieve the lower bias, so it requires more times.

Part 4: Appendix

Link to github repo:

https://github.com/yaxuanw3/IE517_F20_HW6 (https://github.com/yaxuanw3/IE517_F20_HW6)

```
My name is {Yaxuan Wang}
My NetID is: {662869931}
I hereby certify that I have read the University policy on Academic Integrity
and that I am not in violation.
```