Machine Learning in Finance Lab: Week 06

deadline 2022-03-05

- Yu-Ching Liao ycliao3@illinois.edu

Basic Import

In [55]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.image as mpimg

from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn import metrics
from pydotplus import graph_from_dot_data
from sklearn.tree import export_graphviz
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import StratifiedKFold
```

In [7]:
```
cc = pd.read_csv(
    "/Users/yu-chingliao/Library/CloudStorage/GoogleDrive-josephliao0127@gma
    index_col='ID')
cc
```

Out[7]:

ID	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	PAY_0	PAY_2	PAY_3	PAY_4	PAY_
1	20000	2	2	1	24	2	2	-1	-1	
2	120000	2	2	2	26	-1	2	0	0	
3	90000	2	2	2	34	0	0	0	0	
4	50000	2	2	1	37	0	0	0	0	
5	50000	1	2	1	57	-1	0	-1	0	
...	...	...	...	...	...	...	...	...	...	
29996	220000	1	3	1	39	0	0	0	0	
29997	150000	1	3	2	43	-1	-1	-1	-1	
29998	30000	1	2	2	37	4	3	2	-1	
29999	80000	1	3	1	41	1	-1	0	0	
30000	50000	1	2	1	46	0	0	0	0	

30000 rows × 24 columns

In [18]:
```
x = cc.drop("DEFAULT", axis=1).values
y =cc["DEFAULT"].values
```

Random Test-Train Splits

In [77]:
```
RDST = []
BEST_DEPTH = []
BEST_IN_ACC = []
BEST_OUT_ACC = []

#Process on different random states
for rd_st in range(1, 11):
    X_train, X_test, y_train, y_test = train_test_split(x,
                                                        y,
                                                        test_size=0.1,
                                                        random_state=rd_st,
                                                        stratify=y)


    #Modified code from Module 2 starts.
    tree = DecisionTreeClassifier(criterion='gini',
                                  max_depth=None,
                                  random_state=1)
    tree.fit(X_train, y_train)

    y_pred_train = tree.predict(X_train)
    y_pred = tree.predict(X_test)

    out_acc = metrics.accuracy_score(y_test, y_pred)
    in_acc = metrics.accuracy_score(y_train, y_pred_train)
```

```
    depth = tree.get_depth()

    RDST.append(rd_st)
    BEST_DEPTH.append(depth)
    BEST_IN_ACC.append(in_acc)
    BEST_OUT_ACC.append(out_acc)
    #Modified code from Modulo 2 ends.

#Display output
display_df = {
    "Random State": RDST,
    "Best max_depth": BEST_DEPTH,
    "In Sample Scores": BEST_IN_ACC,
    "Out of Sample Scores": BEST_OUT_ACC
}
display_df = pd.DataFrame(display_df)
display_df = display_df.set_index('Random State', drop=True)
display_df = display_df.transpose()
display(display_df)

display_df_2 = {
    "In Sample Scores": [np.mean(BEST_IN_ACC),
                         np.std(BEST_IN_ACC)],
    "Out of Sample Scores": [np.mean(BEST_OUT_ACC),
                           np.std(BEST_OUT_ACC)]
}
display_df_2 = pd.DataFrame(display_df_2, index=['μ', 'σ'])
display_df_2 = display_df_2.transpose()
display(display_df_2)
```

Random State	1	2	3	4	5	6	7
Best max_depth	37.000000	40.000000	45.000000	40.000000	47.000000	50.000000	47.000000
In Sample Scores	0.999333	0.999370	0.999444	0.999407	0.999296	0.999296	0.999296
Out of Sample Scores	0.724333	0.720667	0.721667	0.732000	0.722333	0.710333	0.739000

	μ	σ
In Sample Scores	0.999356	0.000047
Out of Sample Scores	0.722300	0.007929

Cross validation

In [81]:
```
kf = StratifiedKFold(n_splits=10)
tree = DecisionTreeClassifier(criterion='gini', max_depth=None, random_state
cv_scores = cross_val_score(tree, x, y, cv=kf)
```

```
display_df = {"Fold": list(range(1, 11)), "Scores": cv_scores}
display_df = pd.DataFrame(display_df)
display_df = display_df.set_index('Fold', drop=True)
display_df = display_df.transpose()
display(display_df)

display_df_2 = {"Scores": [np.mean(cv_scores), np.std(cv_scores)]}
display_df_2 = pd.DataFrame(display_df_2, index=['μ', 'σ'])
display_df_2 = display_df_2.transpose()
display(display_df_2)
```

Fold	1	2	3	4	5	6	7	8	9	
Scores	0.712667	0.726	0.717333	0.713667	0.718667	0.727333	0.735	0.741	0.738	0.7246

	μ	σ
Scores	0.725433	0.009535

Conclusion

From both of the results, either way provides similar outcome in out-sample accuracy. However, cross-validation provides a more efficient process that we do not have to do tunning by ourselves.

Signing

My name is Yu-Ching Liao
My NetID is: 656724372
I hereby certify that I have read the University policy on Academic Integrity and that I am not in violation.