Lab 3_2

2. Credit card fraud data.

Download the data set at

https://www.kaggle.com/mlg-ulb/creditcardfraud.

This data set has details of 284,807 credit card transactions, some of which are fraudulent. Each transaction is represented by 28 features (scrambled using PCA as a primitive kind of anonymization), and has a corresponding label (1 is fraudulent and 0 is legitimate).

In [1]:

```
%matplotlib inline
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from pandas import read_csv
from matplotlib.pyplot import figure
from sklearn.model_selection import cross_val_score
from sklearn.tree import DecisionTreeClassifier
import graphviz
from sklearn import tree
from sklearn.ensemble import AdaBoostClassifier
from sklearn.model_selection import cross_val_predict
from sklearn.metrics import confusion_matrix
from sklearn.ensemble import RandomForestClassifier
```

In [2]:

```
data = read_csv('creditcard.csv')
data
```

Out[2]:

	Time	V1	V2	V3	V4	V5	V6	V7
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803
2	1.0	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461
3	1.0	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1.247203	0.237609
4	2.0	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.592941
...	...	...	...	...	...	...	...	...
284802	172786.0	-11.881118	10.071785	-9.834783	-2.066656	-5.364473	-2.606837	-4.918215
284803	172787.0	-0.732789	-0.055080	2.035030	-0.738589	0.868229	1.058415	0.024330
284804	172788.0	1.919565	-0.301254	-3.249640	-0.557828	2.630515	3.031260	-0.296827
284805	172788.0	-0.240440	0.530483	0.702510	0.689799	-0.377961	0.623708	-0.686180
284806	172792.0	-0.533413	-0.189733	0.703337	-0.506271	-0.012546	-0.649617	1.577006

284807 rows × 31 columns

(a) How many of the transactions are fraudulent? Why might this be problematic when learning a classifier?

In [3]:

```
df_data = pd.DataFrame(data = data)
num = df_data[df_data['Class'] == 1]['Class'].count()
print('There are', num, 'transactions are fraudulent.')
```

```
There are 492 transactions are fraudulent.
```

The dataset for fraud data is too small compared to the legitimate data. If we are learning a classifier, it might overfit.

(b) Downsample the legitimate transactions to make the data set more balanced.

In [4]:

```
from sklearn.model_selection import train_test_split
fraud_df = df_data[df_data['Class'] == 1]
legit_df = df_data[df_data['Class'] == 0]
legit_data_df = legit_df.sample(n=492, axis=0)
df  = pd.DataFrame(fraud_df)
df  = df.append(legit_data_df)
df
```

Out[4]:

	Time	V1	V2	V3	V4	V5	V6	V7
541	406.0	-2.312227	1.951992	-1.609851	3.997906	-0.522188	-1.426545	-2.537387
623	472.0	-3.043541	-3.157307	1.088463	2.288644	1.359805	-1.064823	0.325574
4920	4462.0	-2.303350	1.759247	-0.359745	2.330243	-0.821628	-0.075788	0.562320
6108	6986.0	-4.397974	1.358367	-2.592844	2.679787	-1.128131	-1.706536	-3.496197
6329	7519.0	1.234235	3.019740	-4.304597	4.732795	3.624201	-1.357746	1.713445
...	...	...	...	...	...	...	...	...
36567	38593.0	-0.325526	-0.296753	2.009558	-1.956363	-0.405342	0.235913	-0.237918
195324	131000.0	-0.860653	0.856452	0.942480	-0.551812	0.749179	-0.035312	0.936558
264528	161472.0	1.871735	-0.739331	-0.746465	-0.034964	0.289016	1.957493	-1.118675
223507	143411.0	-1.187430	-0.347975	0.794399	-0.398793	0.889006	-0.454716	-0.004612
10467	17110.0	-0.973167	1.658315	1.101799	0.340175	-0.350298	-1.189209	0.302029

984 rows × 31 columns

(c) Fit three kinds of classifier to the data:

- decision tree
- boosted decision stumps
- random forest

In each case, use cross-validation to estimate the confusion matrix.

In [5]:

```
inputs = df.values[:, 1:29]
labels = df.values[:, -1]
```

In [6]:

```
clf = DecisionTreeClassifier(random_state=0, criterion='gini')
clf.fit(inputs, labels)
pred = cross_val_predict(clf, inputs, labels, cv=10)
conf_mat = confusion_matrix(labels, pred)
conf_mat
```

Out[6]:

```
array([[443,  49],
       [ 39, 453]])
```

In [7]:

```
clf_a = AdaBoostClassifier(random_state=0)
clf_a.fit(inputs, labels)
pred_a = cross_val_predict(clf_a, inputs, labels, cv=10)
conf_mat_a = confusion_matrix(labels, pred_a)
conf_mat_a
```

Out[7]:

```
array([[466,  26],
       [ 42, 450]])
```

In [8]:

```
clf_r = RandomForestClassifier(random_state=0)
clf_r.fit(inputs, labels)
pred_r = cross_val_predict(clf_r, inputs, labels, cv=10)
conf_mat_r = confusion_matrix(labels, pred_r)
conf_mat_r
```

Out[8]:

```
array([[479,  13],
       [ 48, 444]])
```

In [ ]: