credit_fraud_RandomForest

April 24, 2018

In [42]:
```
import pandas as pd
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier, export_graphviz
from sklearn.model_selection import train_test_split
import numpy as np
import pickle
```

In [33]:
```
df = pd.read_csv('creditcard.csv')
```

In [34]:
```
y = df['Class']
x = df
x = x.drop('Class',axis=1)
x.head()
```

Out[34]:

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V20	V21	V22	V23	V24	V25	V26	V27	V28	Amount
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599	0.098698	0.363787	...	0.251412	-0.018307	0.277838	-0.110474	0.066928	0.128539	-0.189115	0.133558	-0.021053	149.62
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803	0.085102	-0.255425	...	-0.069083	-0.225775	-0.638672	0.101288	-0.339846	0.167170	0.125895	-0.008983	0.014724	2.69
2	1.0	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461	0.247676	-1.514654	...	0.524980	0.247998	0.771679	0.909412	-0.689281	-0.327642	-0.139097	-0.055353	-0.059752	378.66
3	1.0	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1.247203	0.237609	0.377436	-1.387024	...	-0.208038	-0.108300	0.005274	-0.190321	-1.175575	0.647376	-0.221929	0.062723	0.061458	123.50
4	2.0	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.592941	-0.270533	0.817739	...	0.408542	-0.009431	0.798278	-0.137458	0.141267	-0.206010	0.502292	0.219422	0.215153	69.99

[5 rows x 30 columns]

In [35]:
```
x_train, x_test, y_train, y_test = train_test_split(x,y)
```

```
In [45]: rfc = RandomForestClassifier()
         rfc.fit(x_train,y_train)

Out[45]: RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
                     max_depth=None, max_features='auto', max_leaf_nodes=None,
                     min_impurity_decrease=0.0, min_impurity_split=None,
                     min_samples_leaf=1, min_samples_split=2,
                     min_weight_fraction_leaf=0.0, n_estimators=10, n_jobs=1,
                     oob_score=False, random_state=None, verbose=0,
                     warm_start=False)

In [39]: rfc = pickle.dump(rfc,open('RFC_creditcard','wb'))

In [40]: rfc = pickle.load(open('RFC_creditcard','rb'))

In [37]: predicted = rfc.predict(x_test)
         np.mean((predicted-y_test)**2)

Out[37]: 0.0005758265217269177

In [47]: rfc.score(x_test,y_test)

Out[47]: 0.9994803516755147

In [50]: # export_graphviz(rfc.estimators_[0],feature_names=x.columns,filled=True, rounded=True)

/home/multiplexer/anaconda2/envs/py36/lib/python3.6/site-packages/sklearn/tree/export.py:399: De
  DeprecationWarning)
```