Week 2. Running a Random Forest

June 21, 2016

In [1]:
```
#
# Created on Sun Dec 13 21:12:54 2015
#
# @author: ldierker
# Modified by: mcolosso
#
```

In [2]:
```
%matplotlib inline

from pandas import Series, DataFrame
import pandas as pd
import numpy as np
import os
import matplotlib.pylab as plt
from sklearn.cross_validation import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import classification_report
import sklearn.metrics

# Feature Importance
from sklearn import datasets
from sklearn.ensemble import ExtraTreesClassifier

#pd.set_option('display.float_format', lambda x:'%.3f'%x)
```

In [3]:
```
#os.chdir("C:/Users/MColosso/Documents/CURSOS/Wesleyan University/Machine Learning for Data Ana
```

In [4]:
```
#
# Data Engineering and Analysis
#
```

In [5]:
```
#Load the dataset

loans = pd.read_csv("./LendingClub.csv", low_memory = False)

# LendingClub.csv is a dataset taken from The LendingClub (https://www.lendingclub.com/)
# which is a peer-to-peer leading company that directly connects borrowers and potential
# lenders/investors
```

In [6]:
```
#
# Exploring the target column
#

# The target column (label column) of the dataset that we are interested in is called
```

```
# `bad_loans`. In this column **1** means a risky (bad) loan **0** means a safe  loan.
#
# In order to make this more intuitive, we reassign the target to be:
# * **+1** as a safe  loan,
# * **-1** as a risky (bad) loan.
#
# We put this in a new column called `safe_loans`.
```

In [7]:
```
loans['safe_loans'] = loans['bad_loans'].apply(lambda x : +1 if x==0 else -1)
loans.drop('bad_loans', axis = 1, inplace = True)
```

In [8]:
```
# Select features to handle

predictors = ['grade',                    # grade of the loan
              'sub_grade',                # sub-grade of the loan
              'short_emp',                # one year or less of employment
              'emp_length_num',           # number of years of employment
              'home_ownership',           # home_ownership status: own, mortgage or rent
              'dti',                      # debt to income ratio
              'purpose',                  # the purpose of the loan
              'term',                     # the term of the loan
              'last_delinq_none',         # has borrower had a delinquincy
              'last_major_derog_none',    # has borrower had 90 day or worse rating
              'revol_util',               # percent of available credit being used
              'total_rec_late_fee',       # total late fees received to day
             ]

target = 'safe_loans'                   # prediction target (y) (+1 means safe, -1 is risky)

# Extract the predictors and target columns
loans = loans[predictors + [target]]

# Delete rows where any or all of the data are missing
data_clean = loans.dropna()
```

In [9]:
```
# Convert categorical variables into binary variables
# (Categorical features are not, yet, supported by sklearn DecisionTreeClassifier)

data_clean = pd.get_dummies(data_clean, prefix_sep = '=')
```

In [10]:
```
print(data_clean.dtypes)

 (data_clean.describe()).T
```

```
short_emp                   int64
emp_length_num              int64
dti                       float64
last_delinq_none            int64
last_major_derog_none       int64
revol_util                float64
total_rec_late_fee        float64
safe_loans                  int64
grade=A                   float64
grade=B                   float64
grade=C                   float64
```

```
grade=D                         float64
grade=E                         float64
grade=F                         float64
grade=G                         float64
sub_grade=A1                    float64
sub_grade=A2                    float64
sub_grade=A3                    float64
sub_grade=A4                    float64
sub_grade=A5                    float64
sub_grade=B1                    float64
sub_grade=B2                    float64
sub_grade=B3                    float64
sub_grade=B4                    float64
sub_grade=B5                    float64
sub_grade=C1                    float64
sub_grade=C2                    float64
sub_grade=C3                    float64
sub_grade=C4                    float64
sub_grade=C5                    float64
                                 ...
sub_grade=E4                    float64
sub_grade=E5                    float64
sub_grade=F1                    float64
sub_grade=F2                    float64
sub_grade=F3                    float64
sub_grade=F4                    float64
sub_grade=F5                    float64
sub_grade=G1                    float64
sub_grade=G2                    float64
sub_grade=G3                    float64
sub_grade=G4                    float64
sub_grade=G5                    float64
home_ownership=MORTGAGE         float64
home_ownership=OTHER            float64
home_ownership=OWN              float64
home_ownership=RENT             float64
purpose=car                     float64
purpose=credit_card             float64
purpose=debt_consolidation      float64
purpose=home_improvement        float64
purpose=house                   float64
purpose=major_purchase          float64
purpose=medical                 float64
purpose=moving                  float64
purpose=other                   float64
purpose=small_business          float64
purpose=vacation                float64
purpose=wedding                 float64
term= 36 months                 float64
term= 60 months                 float64
dtype: object
```

Out[10]:

```
                               count      mean       std  min   25%   50%  \
         short_emp          122607.0  0.123672  0.329208  0.0  0.00  0.00
```

emp_length_num	122607.0	6.370256	3.736014	0.0	3.00	6.00
dti	122607.0	15.496888	7.497442	0.0	9.88	15.26
last_delinq_none	122607.0	0.588115	0.492177	0.0	0.00	1.00
last_major_derog_none	122607.0	0.873906	0.331957	0.0	1.00	1.00
revol_util	122607.0	53.716307	25.723881	0.0	34.80	55.70
total_rec_late_fee	122607.0	0.742344	5.363268	0.0	0.00	0.00
safe_loans	122607.0	0.622371	0.782726	-1.0	1.00	1.00
grade=A	122607.0	0.181996	0.385843	0.0	0.00	0.00
grade=B	122607.0	0.303180	0.459634	0.0	0.00	0.00
grade=C	122607.0	0.244276	0.429659	0.0	0.00	0.00
grade=D	122607.0	0.156394	0.363230	0.0	0.00	0.00
grade=E	122607.0	0.073324	0.260668	0.0	0.00	0.00
grade=F	122607.0	0.032070	0.176187	0.0	0.00	0.00
grade=G	122607.0	0.008760	0.093183	0.0	0.00	0.00
sub_grade=A1	122607.0	0.024362	0.154172	0.0	0.00	0.00
sub_grade=A2	122607.0	0.027339	0.163071	0.0	0.00	0.00
sub_grade=A3	122607.0	0.032258	0.176684	0.0	0.00	0.00
sub_grade=A4	122607.0	0.048880	0.215617	0.0	0.00	0.00
sub_grade=A5	122607.0	0.049157	0.216197	0.0	0.00	0.00
sub_grade=B1	122607.0	0.047607	0.212935	0.0	0.00	0.00
sub_grade=B2	122607.0	0.057876	0.233510	0.0	0.00	0.00
sub_grade=B3	122607.0	0.073699	0.261281	0.0	0.00	0.00
sub_grade=B4	122607.0	0.067525	0.250930	0.0	0.00	0.00
sub_grade=B5	122607.0	0.056473	0.230834	0.0	0.00	0.00
sub_grade=C1	122607.0	0.057648	0.233077	0.0	0.00	0.00
sub_grade=C2	122607.0	0.054858	0.227704	0.0	0.00	0.00
sub_grade=C3	122607.0	0.046408	0.210369	0.0	0.00	0.00
sub_grade=C4	122607.0	0.044059	0.205228	0.0	0.00	0.00
sub_grade=C5	122607.0	0.041303	0.198990	0.0	0.00	0.00
...	...	...	...	...	...	...
sub_grade=E4	122607.0	0.012895	0.112821	0.0	0.00	0.00
sub_grade=E5	122607.0	0.011092	0.104735	0.0	0.00	0.00
sub_grade=F1	122607.0	0.009013	0.094506	0.0	0.00	0.00
sub_grade=F2	122607.0	0.007585	0.086763	0.0	0.00	0.00
sub_grade=F3	122607.0	0.006280	0.078999	0.0	0.00	0.00
sub_grade=F4	122607.0	0.005130	0.071442	0.0	0.00	0.00
sub_grade=F5	122607.0	0.004062	0.063603	0.0	0.00	0.00
sub_grade=G1	122607.0	0.003018	0.054852	0.0	0.00	0.00
sub_grade=G2	122607.0	0.001966	0.044292	0.0	0.00	0.00
sub_grade=G3	122607.0	0.001362	0.036881	0.0	0.00	0.00
sub_grade=G4	122607.0	0.001240	0.035188	0.0	0.00	0.00
sub_grade=G5	122607.0	0.001174	0.034251	0.0	0.00	0.00
home_ownership=MORTGAGE	122607.0	0.483170	0.499719	0.0	0.00	0.00
home_ownership=OTHER	122607.0	0.001460	0.038182	0.0	0.00	0.00
home_ownership=OWN	122607.0	0.081097	0.272984	0.0	0.00	0.00
home_ownership=RENT	122607.0	0.434274	0.495663	0.0	0.00	0.00
purpose=car	122607.0	0.019371	0.137825	0.0	0.00	0.00
purpose=credit_card	122607.0	0.179843	0.384058	0.0	0.00	0.00
purpose=debt_consolidation	122607.0	0.556518	0.496797	0.0	0.00	1.00
purpose=home_improvement	122607.0	0.061522	0.240286	0.0	0.00	0.00
purpose=house	122607.0	0.008197	0.090165	0.0	0.00	0.00
purpose=major_purchase	122607.0	0.031621	0.174991	0.0	0.00	0.00
purpose=medical	122607.0	0.013107	0.113733	0.0	0.00	0.00
purpose=moving	122607.0	0.009624	0.097630	0.0	0.00	0.00

purpose=other	122607.0	0.074115	0.261959	0.0	0.00	0.00
purpose=small_business	122607.0	0.026622	0.160976	0.0	0.00	0.00
purpose=vacation	122607.0	0.007014	0.083457	0.0	0.00	0.00
purpose=wedding	122607.0	0.012446	0.110867	0.0	0.00	0.00
term= 36 months	122607.0	0.797679	0.401732	0.0	1.00	1.00
term= 60 months	122607.0	0.202321	0.401732	0.0	0.00	0.00

	75%	max
short_emp	0.00	1.00
emp_length_num	11.00	11.00
dti	20.85	39.88
last_delinq_none	1.00	1.00
last_major_derog_none	1.00	1.00
revol_util	74.30	150.70
total_rec_late_fee	0.00	208.82
safe_loans	1.00	1.00
grade=A	0.00	1.00
grade=B	1.00	1.00
grade=C	0.00	1.00
grade=D	0.00	1.00
grade=E	0.00	1.00
grade=F	0.00	1.00
grade=G	0.00	1.00
sub_grade=A1	0.00	1.00
sub_grade=A2	0.00	1.00
sub_grade=A3	0.00	1.00
sub_grade=A4	0.00	1.00
sub_grade=A5	0.00	1.00
sub_grade=B1	0.00	1.00
sub_grade=B2	0.00	1.00
sub_grade=B3	0.00	1.00
sub_grade=B4	0.00	1.00
sub_grade=B5	0.00	1.00
sub_grade=C1	0.00	1.00
sub_grade=C2	0.00	1.00
sub_grade=C3	0.00	1.00
sub_grade=C4	0.00	1.00
sub_grade=C5	0.00	1.00
...	...	...
sub_grade=E4	0.00	1.00
sub_grade=E5	0.00	1.00
sub_grade=F1	0.00	1.00
sub_grade=F2	0.00	1.00
sub_grade=F3	0.00	1.00
sub_grade=F4	0.00	1.00
sub_grade=F5	0.00	1.00
sub_grade=G1	0.00	1.00
sub_grade=G2	0.00	1.00
sub_grade=G3	0.00	1.00
sub_grade=G4	0.00	1.00
sub_grade=G5	0.00	1.00
home_ownership=MORTGAGE	1.00	1.00
home_ownership=OTHER	0.00	1.00
home_ownership=OWN	0.00	1.00

```
home_ownership=RENT          1.00    1.00
purpose=car                   0.00    1.00
purpose=credit_card          0.00    1.00
purpose=debt_consolidation   1.00    1.00
purpose=home_improvement     0.00    1.00
purpose=house                 0.00    1.00
purpose=major_purchase       0.00    1.00
purpose=medical               0.00    1.00
purpose=moving                0.00    1.00
purpose=other                 0.00    1.00
purpose=small_business       0.00    1.00
purpose=vacation              0.00    1.00
purpose=wedding               0.00    1.00
term= 36 months               1.00    1.00
term= 60 months               0.00    1.00

[68 rows x 8 columns]
```

In [11]:
```python
# Extract new features names
features = data_clean.columns.values
features = features[features != target]
```

In [12]:
```python
#
# Modeling and Prediction
#
```

In [13]:
```python
#Split into training and testing sets

predictors = data_clean[features]

targets = data_clean.safe_loans

pred_train, pred_test, tar_train, tar_test = train_test_split(predictors, targets,
                                                              test_size = .4)

print('pred_train.shape', pred_train.shape)
print('pred_test.shape',  pred_test.shape)
print('tar_train.shape',  tar_train.shape)
print('tar_test.shape',   tar_test.shape)
```

```
pred_train.shape (73564, 67)
pred_test.shape (49043, 67)
tar_train.shape (73564,)
tar_test.shape (49043,)
```

In [14]:
```python
#Build model on training data
from sklearn.ensemble import RandomForestClassifier

classifier = RandomForestClassifier(n_estimators = 25)
classifier = classifier.fit(pred_train, tar_train)

predictions = classifier.predict(pred_test)

conf_matrix = sklearn.metrics.confusion_matrix(tar_test, predictions)

print(conf_matrix)
```

```
[[ 1200  7957]
 [ 2068 37818]]
```

In [15]:
```
sklearn.metrics.accuracy_score(tar_test, predictions)
```

Out[15]:
```
0.79558754562322864
```

In [16]:
```
# fit an Extra Trees model to the data
model = ExtraTreesClassifier()
model.fit(pred_train, tar_train)
# display the relative importance of each attribute
print(model.feature_importances_)
```

```
[ 0.0116846   0.13741726  0.29752994  0.02180671  0.01403449  0.28940768
  0.03293531  0.00951938  0.0037876   0.00329254  0.00387515  0.00546364
  0.00417739  0.00154924  0.00084011  0.00069246  0.00081152  0.00096394
  0.00116574  0.00167741  0.00205909  0.00237266  0.00246947  0.00251691
  0.00305698  0.00325398  0.00302473  0.00332757  0.00274992  0.00220827
  0.00224134  0.00220883  0.00217347  0.00213623  0.00171983  0.00202893
  0.00204825  0.0020539   0.00177719  0.00121543  0.00119387  0.00116458
  0.00113021  0.00095636  0.00060222  0.00054441  0.00046878  0.00038406
  0.00040649  0.00745245  0.00060827  0.00606502  0.00744603  0.00355398
  0.00929081  0.01264373  0.00652372  0.00243849  0.00420151  0.0037273
  0.00321748  0.00838174  0.00601524  0.00236188  0.00309275  0.00558167
  0.01127186]
```

In [17]:
```
# Show more important features
more_important_features = list()
predictors_list = list(predictors.columns.values)
idx = 0
for imp in model.feature_importances_:
    if imp >= 0.1:
        more_important_features.append(predictors_list[idx])
    idx += 1
print('More important features:', more_important_features)
```

```
More important features: ['emp_length_num', 'dti', 'revol_util']
```

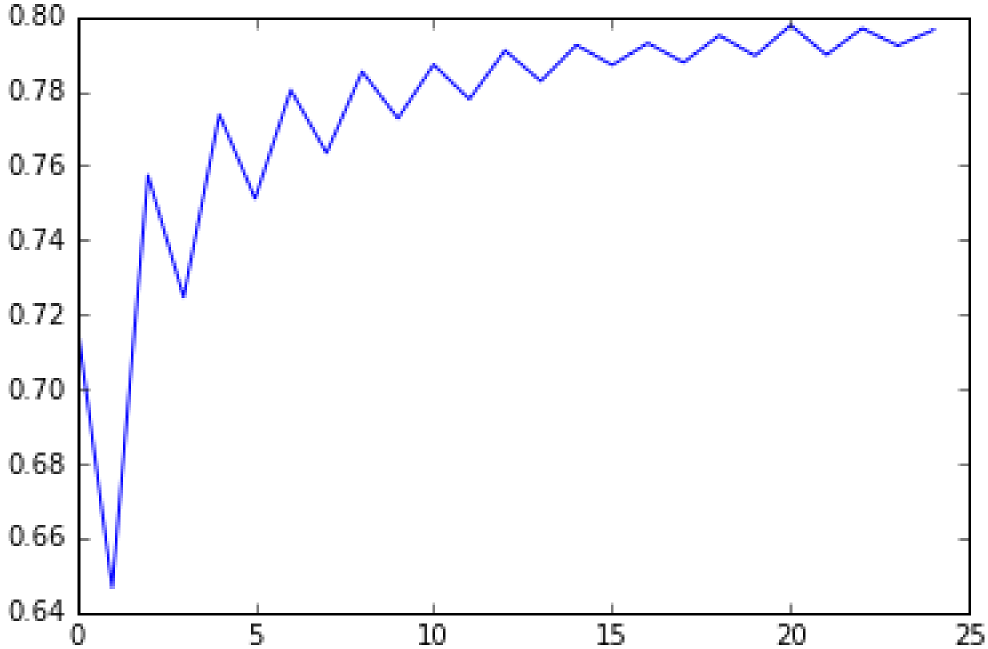
In [18]:
```
#
# Running a different number of trees and see the effect
# of that on the accuracy of the prediction
#
```

In [19]:
```
trees = range(25)
accuracy = np.zeros(25)

for idx in range(len(trees)):
   classifier = RandomForestClassifier(n_estimators = idx + 1)
   classifier = classifier.fit(pred_train,tar_train)
   predictions = classifier.predict(pred_test)
   accuracy[idx] = sklearn.metrics.accuracy_score(tar_test, predictions)

plt.cla()  # Clear axis
plt.plot(trees, accuracy)
```

Out[19]:
```
[<matplotlib.lines.Line2D at 0x18bccd1e3c8>]
```




```
In [ ]:
```