In [1]:
```
import sklearn.externals.joblib as joblib
import sklearn.metrics
import sklearn.externals.joblib as joblib
import numpy as np
import pandas as pd
import seaborn as sns

from sklearn.preprocessing import Normalizer, StandardScaler
from sklearn.decomposition import PCA

from MulticoreTSNE import MulticoreTSNE as TSNE
```

In [2]:
```
data = pd.read_csv('../data/dataset.csv', header=None)
```

In [3]:
```
data.head(5)
```

Out[3]:

	0	1	2	3	4	5	6	7	8	9	...	477	478	479	480	481	482	483	4
0	pe-malicious	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.385031	0.60	0
1	pe-malicious	0.0	0.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.695652	0.20	0
2	pe-malicious	0.0	0.0	0.0	0.0	1.0	0.0	0.0	1.0	1.0	...	0.0	0.0	0.0	0.0	0.0	0.163088	1.00	1
3	pe-malicious	0.0	1.0	0.0	0.0	1.0	1.0	1.0	1.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.925532	0.25	0
4	pe-malicious	0.0	0.0	0.0	0.0	1.0	0.0	0.0	1.0	1.0	...	0.0	0.0	0.0	0.0	0.0	0.220399	1.00	1

5 rows × 487 columns

In [4]:
```
desc = data.describe()
desc
```

Out[4]:

	1	2	3	4	5	6	
count	199970.000000	199970.000000	199970.0	199970.000000	199970.000000	199970.000000	199970
mean	0.231620	0.266580	0.0	0.071796	0.951718	0.258664	0
std	0.421868	0.442172	0.0	0.258150	0.214363	0.437902	0
min	0.000000	0.000000	0.0	0.000000	0.000000	0.000000	0
25%	0.000000	0.000000	0.0	0.000000	1.000000	0.000000	0
50%	0.000000	0.000000	0.0	0.000000	1.000000	0.000000	0
75%	0.000000	1.000000	0.0	0.000000	1.000000	1.000000	1
max	1.000000	1.000000	0.0	1.000000	1.000000	1.000000	1

8 rows × 486 columns

In [5]:
```
desc.T[desc.T['min'] == desc.T['max']]
```

Out[5]:

	count	mean	std	min	25%	50%	75%	max
3	199970.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

In [ ]:
```
sample = data.sample(frac=0.02)

y = sample.loc[:,0].values
X = sample.loc[:,1:].values
X = Normalizer().fit_transform(X)
```

In [7]:
```
tsne = TSNE(
    n_components=2,
    n_jobs=-1
).fit_transform(X)
```

In [8]:
```
plot_df = pd.concat(
    [pd.DataFrame(y, columns=['class']), pd.DataFrame(tsne, columns=['x
    axis=1
)
plot_df.head()
```

Out[8]:

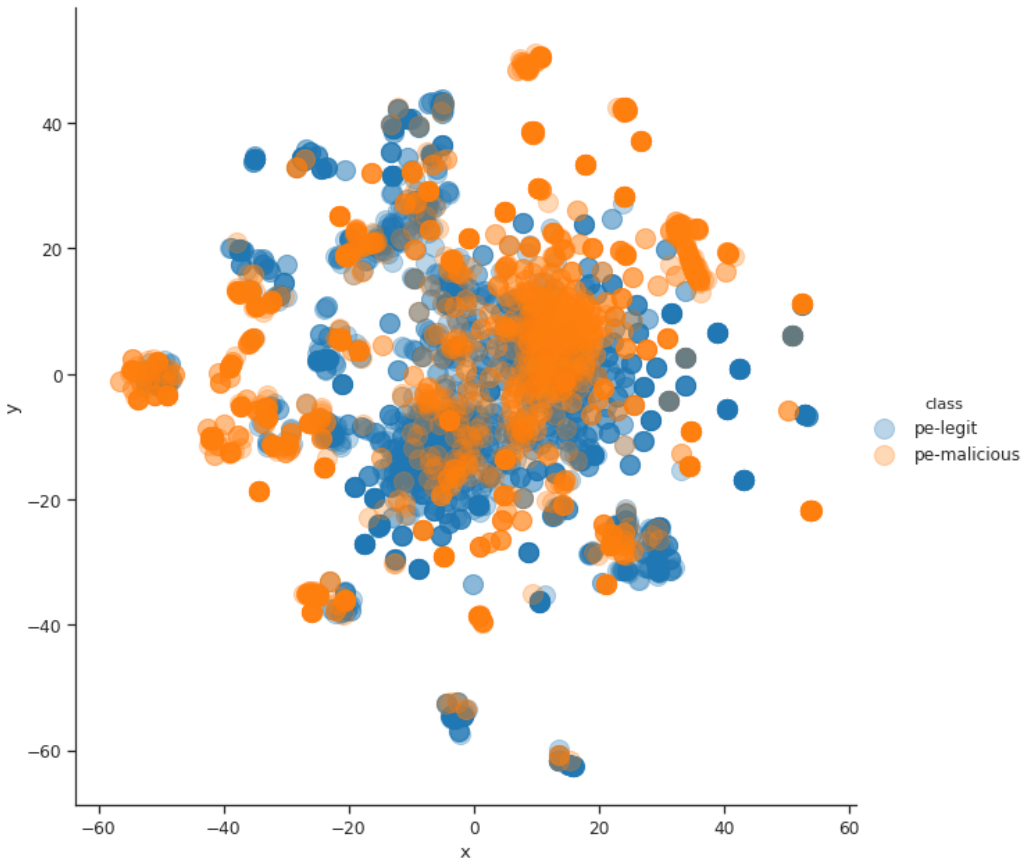
	class	x	y
0	pe-legit	-7.344491	43.114080
1	pe-malicious	25.613658	-4.990510
2	pe-legit	9.936892	-15.312107
3	pe-malicious	21.753220	-27.393085
4	pe-legit	9.061579	4.553387

In [9]:
```
# Set style of scatterplot
sns.set_context("notebook", font_scale=1.1)
sns.set_style("ticks")

# Create scatterplot of dataframe
sns.lmplot(x='x',
           y='y',
           data=plot_df,
           fit_reg=False,
           legend=True,
           size=9,
           hue='class',
           scatter_kws={"s":200, "alpha":0.3})
```

```
/usr/local/lib/python2.7/site-packages/seaborn/regression.py:546: User
Warning: The `size` paramter has been renamed to `height`; please upda
te your code.
  warnings.warn(msg, UserWarning)
```

Out[9]:
```
<seaborn.axisgrid.FacetGrid at 0x7f4a19337810>
```



In [ ]:
```
```

In [10]:
```
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
```

In [11]:
```
model = LogisticRegression()#C=0.6,dual=True)
```

In [12]:
```
sample = data.sample(frac=1.0)
y = sample.loc[:,0].values
X = sample.loc[:,1:].values
X = Normalizer().fit_transform(X)
```

In [13]:
```
X_train, X_test, y_train, y_test = dupa = train_test_split(X,y, train_s
```

```
/usr/local/lib/python2.7/site-packages/sklearn/model_selection/_split.
py:2179: FutureWarning: From version 0.21, test_size will always compl
ement train_size unless both are specified.
  FutureWarning)
```

In [14]:
```
model.fit(X_train, y_train)
```

```
/usr/local/lib/python2.7/site-packages/sklearn/linear_model/logistic.p
y:433: FutureWarning: Default solver will be changed to 'lbfgs' in
0.22. Specify a solver to silence this warning.
  FutureWarning)
```

Out[14]:
```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercep
t=True,
          intercept_scaling=1, max_iter=100, multi_class='warn',
          n_jobs=None, penalty='l2', random_state=None, solver='warn',
          tol=0.0001, verbose=0, warm_start=False)
```

In [15]:
```
model.score(X_test, y_test)
```

Out[15]:
```
0.848452267840176
```

In [ ]:
```

```

In [16]:
```
sample = data.sample(frac=1.0)
y = sample.loc[:,0].values
X = sample.loc[:,1:].values

pca = PCA(n_components=10).fit_transform(X)
X = np.hstack((X, pca))
X = StandardScaler().fit_transform(X)
```

In [ ]:
```

```

In [ ]:
```

```

In [17]:
```
from xgboost import XGBClassifier, plot_importance
```

In [18]:

```
model = XGBClassifier(colsample_bytree=0.7, subsample=0.7, max_depth=20
model.fit(X_train, y_train, eval_set=[(X_test, y_test)])
```

```
[0]     validation_0-error:0.044107     validation_0-auc:0.978356
[1]     validation_0-error:0.037181     validation_0-auc:0.987783
[2]     validation_0-error:0.03098      validation_0-auc:0.990969
[3]     validation_0-error:0.029679     validation_0-auc:0.991741
[4]     validation_0-error:0.028804     validation_0-auc:0.992387
[5]     validation_0-error:0.027704     validation_0-auc:0.99324
[6]     validation_0-error:0.026954     validation_0-auc:0.993833
[7]     validation_0-error:0.025979     validation_0-auc:0.994255
[8]     validation_0-error:0.025754     validation_0-auc:0.994327
[9]     validation_0-error:0.024979     validation_0-auc:0.994802
[10]    validation_0-error:0.024954     validation_0-auc:0.995192
[11]    validation_0-error:0.024504     validation_0-auc:0.995275
[12]    validation_0-error:0.023779     validation_0-auc:0.995595
[13]    validation_0-error:0.023153     validation_0-auc:0.995835
[14]    validation_0-error:0.022478     validation_0-auc:0.996094
[15]    validation_0-error:0.021978     validation_0-auc:0.99619
[16]    validation_0-error:0.021553     validation_0-auc:0.996254
[17]    validation_0-error:0.021078     validation_0-auc:0.996367
[18]    validation_0-error:0.020803     validation_0-auc:0.996502
[19]    validation_0-error:0.020303     validation_0-auc:0.996605
[20]    validation_0-error:0.020128     validation_0-auc:0.996709
[21]    validation_0-error:0.019728     validation_0-auc:0.996813
[22]    validation_0-error:0.019503     validation_0-auc:0.996876
[23]    validation_0-error:0.019653     validation_0-auc:0.99693
[24]    validation_0-error:0.019378     validation_0-auc:0.996988
[25]    validation_0-error:0.018778     validation_0-auc:0.997077
[26]    validation_0-error:0.018878     validation_0-auc:0.997129
[27]    validation_0-error:0.018378     validation_0-auc:0.99715
[28]    validation_0-error:0.017928     validation_0-auc:0.997198
[29]    validation_0-error:0.017978     validation_0-auc:0.997232
[30]    validation_0-error:0.017853     validation_0-auc:0.997282
[31]    validation_0-error:0.017853     validation_0-auc:0.997285
[32]    validation_0-error:0.017603     validation_0-auc:0.997346
[33]    validation_0-error:0.017778     validation_0-auc:0.99737
[34]    validation_0-error:0.017678     validation_0-auc:0.997407
[35]    validation_0-error:0.017428     validation_0-auc:0.997412
[36]    validation_0-error:0.017328     validation_0-auc:0.997434
[37]    validation_0-error:0.017053     validation_0-auc:0.99747
[38]    validation_0-error:0.017078     validation_0-auc:0.997504
[39]    validation_0-error:0.016878     validation_0-auc:0.99752
[40]    validation_0-error:0.016602     validation_0-auc:0.997531
[41]    validation_0-error:0.016502     validation_0-auc:0.997581
[42]    validation_0-error:0.016252     validation_0-auc:0.997604
[43]    validation_0-error:0.016427     validation_0-auc:0.997622
[44]    validation_0-error:0.016352     validation_0-auc:0.997636
[45]    validation_0-error:0.016127     validation_0-auc:0.99768
[46]    validation_0-error:0.016102     validation_0-auc:0.997701
[47]    validation_0-error:0.016052     validation_0-auc:0.997721
[48]    validation_0-error:0.015752     validation_0-auc:0.997733
[49]    validation_0-error:0.015777     validation_0-auc:0.997755
[50]    validation_0-error:0.015777     validation_0-auc:0.997778
[51]    validation_0-error:0.015577     validation_0-auc:0.997789
[52]    validation_0-error:0.015477     validation_0-auc:0.997806
[53]    validation_0-error:0.015402     validation_0-auc:0.997813
[54]    validation_0-error:0.015452     validation_0-auc:0.997825
[55]    validation_0-error:0.015577     validation_0-auc:0.997838
[56]    validation_0-error:0.015452     validation_0-auc:0.997856
[57]    validation_0-error:0.015352     validation_0-auc:0.997861
[58]    validation_0-error:0.015352     validation_0-auc:0.997868
[59]    validation_0-error:0.015177     validation_0-auc:0.997889
[60]    validation_0-error:0.015227     validation_0-auc:0.997896
[61]    validation_0-error:0.014927     validation_0-auc:0.997919
[62]    validation_0-error:0.014952     validation_0-auc:0.997908
[63]    validation_0-error:0.014827     validation_0-auc:0.997921
[64]    validation_0-error:0.014702     validation_0-auc:0.99793
[65]    validation_0-error:0.014652     validation_0-auc:0.997938
[66]    validation_0-error:0.014752     validation_0-auc:0.997949
```

In [19]:
```
model.score(X_test, y_test)
```

Out[19]: 0.9868730309546432

In [20]:
```
model.score(X_test, y_test)
```

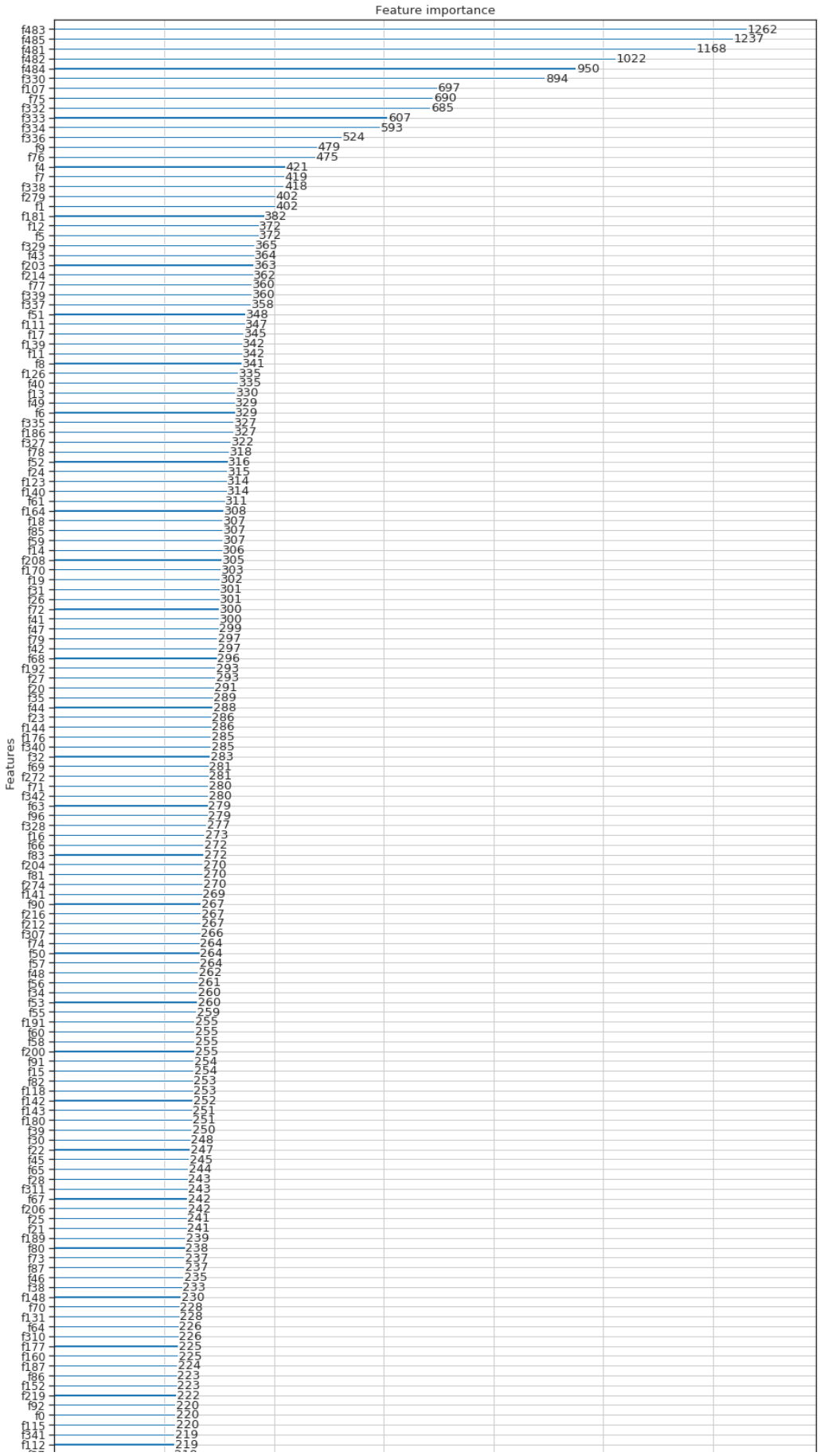
Out[20]: 0.9868730309546432

In [ ]:
```

```

In [26]:
```
from matplotlib import pyplot as plt
fig, ax = plt.subplots(figsize=(15, 30))
plot_importance(model, ax=ax, max_num_features=150, )
```

Out[26]: <matplotlib.axes._subplots.AxesSubplot at 0x7f496408ed90>



In [27]:
```
sample = data.sample(frac=1.0)
y = sample.loc[:,0].values
X = sample.loc[:,1:].values

pca = PCA(n_components=10).fit_transform(X)
X = np.hstack((X, pca))
X = StandardScaler().fit_transform(X)
```

In [28]:
```
X_train, X_test, y_train, y_test = dupa = train_test_split(X,y, test_si
```

In [29]:
```
model = XGBClassifier(colsample_bytree=0.5, subsample=0.5, max_depth=20
model.fit(X_train, y_train, eval_set=[(X_test, y_test)])
```

```
[0]    validation_0-error:0.050508    validation_0-auc:0.971135
[1]    validation_0-error:0.041106    validation_0-auc:0.98659
[2]    validation_0-error:0.034105    validation_0-auc:0.989307
[3]    validation_0-error:0.031055    validation_0-auc:0.991887
[4]    validation_0-error:0.028379    validation_0-auc:0.992787
[5]    validation_0-error:0.027079    validation_0-auc:0.99369
[6]    validation_0-error:0.026304    validation_0-auc:0.994264
[7]    validation_0-error:0.025579    validation_0-auc:0.994573
[8]    validation_0-error:0.025204    validation_0-auc:0.994948
[9]    validation_0-error:0.024804    validation_0-auc:0.995164
[10]   validation_0-error:0.024404    validation_0-auc:0.995489
[11]   validation_0-error:0.023379    validation_0-auc:0.99578
[12]   validation_0-error:0.022953    validation_0-auc:0.996026
[13]   validation_0-error:0.023178    validation_0-auc:0.996131
[14]   validation_0-error:0.022553    validation_0-auc:0.996369
[15]   validation_0-error:0.022403    validation_0-auc:0.996494
[16]   validation_0-error:0.021853    validation_0-auc:0.996662
[17]   validation_0-error:0.021628    validation_0-auc:0.996776
[18]   validation_0-error:0.021328    validation_0-auc:0.996894
[19]   validation_0-error:0.020978    validation_0-auc:0.996991
[20]   validation_0-error:0.020753    validation_0-auc:0.997093
[21]   validation_0-error:0.020378    validation_0-auc:0.997141
[22]   validation_0-error:0.020528    validation_0-auc:0.997232
[23]   validation_0-error:0.020178    validation_0-auc:0.997271
[24]   validation_0-error:0.019978    validation_0-auc:0.99733
[25]   validation_0-error:0.019528    validation_0-auc:0.997379
[26]   validation_0-error:0.019578    validation_0-auc:0.997435
[27]   validation_0-error:0.019203    validation_0-auc:0.997458
[28]   validation_0-error:0.019078    validation_0-auc:0.997479
[29]   validation_0-error:0.018628    validation_0-auc:0.997503
[30]   validation_0-error:0.018303    validation_0-auc:0.997537
[31]   validation_0-error:0.018228    validation_0-auc:0.997568
[32]   validation_0-error:0.018078    validation_0-auc:0.997603
[33]   validation_0-error:0.018078    validation_0-auc:0.997612
[34]   validation_0-error:0.017953    validation_0-auc:0.997635
[35]   validation_0-error:0.017903    validation_0-auc:0.997664
[36]   validation_0-error:0.017803    validation_0-auc:0.997693
[37]   validation_0-error:0.017553    validation_0-auc:0.997731
[38]   validation_0-error:0.017603    validation_0-auc:0.997759
[39]   validation_0-error:0.017203    validation_0-auc:0.997791
[40]   validation_0-error:0.017328    validation_0-auc:0.997811
[41]   validation_0-error:0.017028    validation_0-auc:0.997844
[42]   validation_0-error:0.017253    validation_0-auc:0.997891
[43]   validation_0-error:0.017053    validation_0-auc:0.99792
[44]   validation_0-error:0.016778    validation_0-auc:0.997923
[45]   validation_0-error:0.016678    validation_0-auc:0.997968
[46]   validation_0-error:0.016753    validation_0-auc:0.99798
[47]   validation_0-error:0.016703    validation_0-auc:0.997991
[48]   validation_0-error:0.016778    validation_0-auc:0.997986
[49]   validation_0-error:0.016602    validation_0-auc:0.997987
[50]   validation_0-error:0.016327    validation_0-auc:0.998004
[51]   validation_0-error:0.016327    validation_0-auc:0.998017
[52]   validation_0-error:0.016527    validation_0-auc:0.998024
[53]   validation_0-error:0.016277    validation_0-auc:0.998044
[54]   validation_0-error:0.016302    validation_0-auc:0.99806
[55]   validation_0-error:0.016377    validation_0-auc:0.99807
[56]   validation_0-error:0.016227    validation_0-auc:0.998079
[57]   validation_0-error:0.016227    validation_0-auc:0.998102
[58]   validation_0-error:0.016177    validation_0-auc:0.998105
[59]   validation_0-error:0.015952    validation_0-auc:0.998109
[60]   validation_0-error:0.015977    validation_0-auc:0.998121
[61]   validation_0-error:0.016002    validation_0-auc:0.998137
[62]   validation_0-error:0.015902    validation_0-auc:0.998155
[63]   validation_0-error:0.015727    validation_0-auc:0.998158
[64]   validation_0-error:0.015927    validation_0-auc:0.998161
[65]   validation_0-error:0.015927    validation_0-auc:0.998166
[66]   validation_0-error:0.015827    validation_0-auc:0.998186
```

In [30]:
```
model.score(X_test, y_test)
```

Out[30]:
```
0.9852227834175127
```

In [ ]:
```

```