In [1]:
```
#Import libraries:
import pandas as pd
import numpy as np
import xgboost as xgb
from xgboost.sklearn import XGBClassifier
from sklearn import metrics   #Additional scklearn functions
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import GridSearchCV   #Perforing grid search

import matplotlib.pylab as plt
%matplotlib inline
from matplotlib.pylab import rcParams
rcParams['figure.figsize'] = 12, 4
```

In [2]:
```
train = pd.read_csv('UNSW_NB15_ohe_training_set.csv')
test = pd.read_csv('UNSW_NB15_ohe_testing_set.csv')
train.shape, test.shape
```

Out[2]: ((175341, 201), (82332, 201))

In [3]:
```
train.head(175341)
```

Out[3]:

	id	dur	spkts	dpkts	sbytes	dbytes	rate	sttl	dttl	sload	...	state_ECO	state_FIN	state_INT	sta
0	1	0.121478	6	4	258	172	74.087490	252	254	1.415894e+04	...	0	1	0	
1	2	0.649902	14	38	734	42014	78.473372	62	252	8.395112e+03	...	0	1	0	
2	3	1.623129	8	16	364	13186	14.170161	62	252	1.572272e+03	...	0	1	0	
3	4	1.681642	12	12	628	770	13.677108	62	252	2.740179e+03	...	0	1	0	
4	5	0.449454	10	6	534	268	33.373826	254	252	8.561499e+03	...	0	1	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
175336	175337	0.000009	2	0	114	0	111111.107200	254	0	5.066666e+07	...	0	0	1	
175337	175338	0.505762	10	8	620	354	33.612649	254	252	8.826286e+03	...	0	1	0	
175338	175339	0.000009	2	0	114	0	111111.107200	254	0	5.066666e+07	...	0	0	1	
175339	175340	0.000009	2	0	114	0	111111.107200	254	0	5.066666e+07	...	0	0	1	
175340	175341	0.000009	2	0	114	0	111111.107200	254	0	5.066666e+07	...	0	0	1	

175341 rows × 201 columns

In [4]:
```
test.head(82332)
```

Out[4]:

	id	dur	spkts	dpkts	sbytes	dbytes	rate	sttl	dttl	sload	...	state_ECO	state_FIN	state_INT	state
0	1	0.000011	2	0	496	0	90909.090200	254	0	1.803636e+08	...	0	0	1	
1	2	0.000008	2	0	1762	0	125000.000300	254	0	8.810000e+08	...	0	0	1	
2	3	0.000005	2	0	1068	0	200000.005100	254	0	8.544000e+08	...	0	0	1	
3	4	0.000006	2	0	900	0	166666.660800	254	0	6.000000e+08	...	0	0	1	
4	5	0.000010	2	0	2126	0	100000.002500	254	0	8.504000e+08	...	0	0	1	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
82327	82328	0.000005	2	0	104	0	200000.005100	254	0	8.320000e+07	...	0	0	1	
82328	82329	1.106101	20	8	18062	354	24.410067	254	252	1.241044e+05	...	0	1	0	
82329	82330	0.000000	1	0	46	0	0.000000	0	0	0.000000e+00	...	0	0	1	
82330	82331	0.000000	1	0	46	0	0.000000	0	0	0.000000e+00	...	0	0	1	
82331	82332	0.000009	2	0	104	0	111111.107200	254	0	4.622222e+07	...	0	0	1	

82332 rows × 201 columns

In [5]:
```
target = 'label'
IDcol = 'id'
ACcol = 'attack_cat'
train['label'].value_counts()
```

Out[5]:
```
1    119341
0     56000
Name: label, dtype: int64
```

In [6]:
```
test['label'].value_counts()
```

Out[6]:
```
1    45332
0    37000
Name: label, dtype: int64
```

In [7]:
```
def modelfit(alg, dtrain, dtest, predictors,useTrainCV=True, cv_folds=5, early_stopping_rounds=50):

    if useTrainCV:
        xgb_param = alg.get_xgb_params()
        xgtrain = xgb.DMatrix(dtrain[predictors].values, label=dtrain[target].values)
        xgtest = xgb.DMatrix(dtest[predictors].values)
        cvresult = xgb.cv(xgb_param, xgtrain, num_boost_round=alg.get_params()['n_estimators'], nfold=cv_folds,
            metrics='auc', early_stopping_rounds=early_stopping_rounds)
        n_estimators=cvresult.shape[0]
        alg.set_params(n_estimators=cvresult.shape[0])
        print(cvresult)
        print(n_estimators)

    #Fit the algorithm on the data
    alg.fit(dtrain[predictors], dtrain['label'],eval_metric='auc')

    #Predict training set:
    dtrain_predictions = alg.predict(dtrain[predictors])
    dtrain_predprob = alg.predict_proba(dtrain[predictors])[:,1]
    print(dtrain_predictions)
    print(dtrain_predprob )

    #Print model report:
    print ("\nModel Report")
    print ("Accuracy : %.4g" % metrics.accuracy_score(dtrain['label'].values, dtrain_predictions))
    print ("AUC Score (Train): %f" % metrics.roc_auc_score(dtrain['label'], dtrain_predprob))

#     Predict on testing data:
    dtest_predprob = alg.predict_proba(dtest[predictors])[:,1]
    print ('AUC Score (Test): %f' % metrics.roc_auc_score(dtest['label'], dtest_predprob))

    feat_imp = pd.Series(alg.get_booster().get_fscore()).sort_values(ascending=False)
    feat_imp.plot(kind='bar', title='Feature Importances')
    plt.ylabel('Feature Importance Score')
```

In [8]:
```
predictors = [x for x in train.columns if x not in [target, IDcol, ACcol]]
xgb1 = XGBClassifier(
        learning_rate =0.1,
        n_estimators=1484,
        max_depth=4,
        min_child_weight=12,
        gamma=0.3,
        alpha = 0.05,
        subsample=0.6,
        colsample_bytree=0.6,
        objective= 'binary:logistic',
        nthread=4,
        scale_pos_weight=1,
        seed=27)
modelfit(xgb1, train, test, predictors)
```

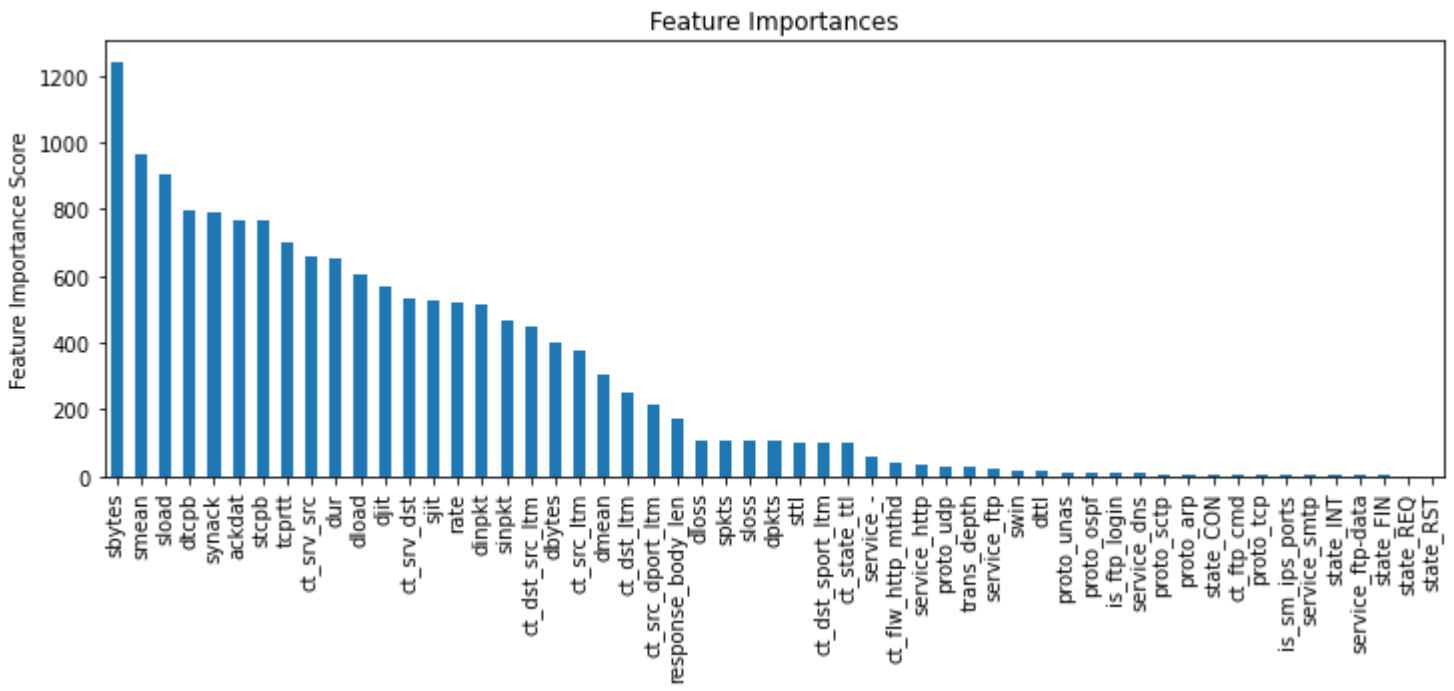
```
      train-auc-mean  train-auc-std  test-auc-mean  test-auc-std
0           0.954683       0.006009       0.954162      0.005136
1           0.974192       0.004496       0.973776      0.005485
2           0.979690       0.002150       0.979447      0.002692
3           0.981730       0.001325       0.981518      0.002134
4           0.982909       0.001122       0.982751      0.001769
...              ...            ...            ...           ...
1479        0.996959       0.000135       0.993515      0.000336
1480        0.996959       0.000136       0.993515      0.000337
1481        0.996961       0.000136       0.993515      0.000337
1482        0.996962       0.000136       0.993516      0.000334
1483        0.996963       0.000137       0.993516      0.000334

[1484 rows x 4 columns]
1484

C:\Users\admin\anaconda3\lib\site-packages\xgboost\sklearn.py:1224: UserWarning: The use of label encoder in X
GBClassifier is deprecated and will be removed in a future release. To remove this warning, do the following:
1) Pass option use_label_encoder=False when constructing XGBClassifier object; and 2) Encode your labels (y) a
s integers starting with 0, i.e. 0, 1, 2, ..., [num_class - 1].
  warnings.warn(label_encoder_deprecation_msg, UserWarning)

[1 0 1 ... 1 1 1]
[0.56998926 0.03172128 0.7467671  ... 0.9998087  0.99999225 0.99999225]

Model Report
Accuracy : 0.9723
AUC Score (Train): 0.996902
AUC Score (Test): 0.982351
```



In [10]:
```
# min_child_weight and max_depth
param_test1 = {
 'max_depth':range(3,10,2),
 'min_child_weight':range(1,6,2)
}
gsearch1 = GridSearchCV(estimator = XGBClassifier( learning_rate =0.1, n_estimators=1484, max_depth=5,
 min_child_weight=1, gamma=0, subsample=0.8, colsample_bytree=0.8,
 objective= 'binary:logistic', nthread=4, scale_pos_weight=1, seed=27),
 param_grid = param_test1, scoring='roc_auc',n_jobs=4, cv=5)
gsearch1.fit(train[predictors],train[target])
gsearch1.cv_results_, gsearch1.best_params_, gsearch1.best_score_
```

```
C:\Users\admin\anaconda3\lib\site-packages\xgboost\sklearn.py:1224: UserWarning: The use of label encoder in X
GBClassifier is deprecated and will be removed in a future release. To remove this warning, do the following:
1) Pass option use_label_encoder=False when constructing XGBClassifier object; and 2) Encode your labels (y) a
s integers starting with 0, i.e. 0, 1, 2, ..., [num_class - 1].
  warnings.warn(label_encoder_deprecation_msg, UserWarning)

[09:39:16] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.5.1/src/learner.cc:1115: Starting
in XGBoost 1.3.0, the default evaluation metric used with the objective 'binary:logistic' was changed from 'er
ror' to 'logloss'. Explicitly set eval_metric if you'd like to restore the old behavior.
```

Out[10]:
```
({'mean_fit_time': array([ 729.07954907,  721.28482447,  705.19047151, 1174.94892302,
        1136.23560157, 1093.03463545, 1663.33019209, 1565.91993876,
        1505.46529331, 2136.15486083, 1992.03475027, 1737.89494276]),
  'std_fit_time': array([  2.45632772,   6.09019432,   9.64368213,  28.52239646,
          7.20569349,  31.90376791,  48.27968294,  18.33054567,
         62.32298757,  66.45935853,  31.40903026, 227.89799608]),
  'mean_score_time': array([0.67495375, 0.71870208, 0.66245537, 1.07492681, 1.10304961,
        1.10929971, 1.58114166, 1.50927229, 1.65926151, 1.87799745,
        1.91549435, 1.7361239 ]),
 'std_score_time': array([0.05537125, 0.08148442, 0.03775702, 0.06874522, 0.02724108,
        0.04940706, 0.10976865, 0.05978014, 0.21063835, 0.12475602,
        0.0830285 , 0.26817182]),
 'param_max_depth': masked_array(data=[3, 3, 3, 5, 5, 5, 7, 7, 7, 9, 9, 9],
              mask=[False, False, False, False, False, False, False, False,
                    False, False, False, False],
        fill_value='?',
             dtype=object),
 'param_min_child_weight': masked_array(data=[1, 3, 5, 1, 3, 5, 1, 3, 5, 1, 3, 5],
              mask=[False, False, False, False, False, False, False, False,
                    False, False, False, False],
        fill_value='?',
             dtype=object),
 'params': [{'max_depth': 3, 'min_child_weight': 1},
  {'max_depth': 3, 'min_child_weight': 3},
  {'max_depth': 3, 'min_child_weight': 5},
  {'max_depth': 5, 'min_child_weight': 1},
  {'max_depth': 5, 'min_child_weight': 3},
  {'max_depth': 5, 'min_child_weight': 5},
  {'max_depth': 7, 'min_child_weight': 1},
  {'max_depth': 7, 'min_child_weight': 3},
  {'max_depth': 7, 'min_child_weight': 5},
  {'max_depth': 9, 'min_child_weight': 1},
  {'max_depth': 9, 'min_child_weight': 3},
  {'max_depth': 9, 'min_child_weight': 5}],
 'split0_test_score': array([0.99554244, 0.99508918, 0.99514837, 0.99527759, 0.99535158,
        0.99541283, 0.99536256, 0.995325  , 0.99546396, 0.9954506 ,
        0.99544213, 0.99533251]),
 'split1_test_score': array([0.99847559, 0.9985761 , 0.99847211, 0.99854981, 0.99859067,
        0.99855776, 0.99847428, 0.99849257, 0.99855198, 0.99854248,
        0.99854064, 0.99849493]),
 'split2_test_score': array([0.99925066, 0.99923841, 0.99924094, 0.99921311, 0.99924044,
        0.99920883, 0.99919851, 0.99916375, 0.99916702, 0.99916903,
        0.99916858, 0.99917347]),
 'split3_test_score': array([0.99261544, 0.99251946, 0.99249558, 0.99187965, 0.9917824 ,
        0.9918384 , 0.9911984 , 0.99129559, 0.99131718, 0.99047615,
        0.99065427, 0.99081135]),
 'split4_test_score': array([0.98078173, 0.98050364, 0.98064568, 0.9779908 , 0.97743334,
        0.97851656, 0.97928449, 0.97856511, 0.97781686, 0.97841752,
        0.97786965, 0.97793123]),
 'mean_test_score': array([0.99333317, 0.99318536, 0.99320054, 0.99258219, 0.99247969,
        0.99270688, 0.99270365, 0.9925684 , 0.9924634 , 0.99241115,
        0.99233505, 0.9923487 ]),
 'std_test_score': array([0.00669919, 0.00679107, 0.00672594, 0.00774918, 0.0079772 ,
        0.0075624 , 0.00727864, 0.00753532, 0.00783387, 0.00764286,
        0.00783496, 0.00779081]),
 'rank_test_score': array([ 1,  3,  2,  6,  8,  4,  5,  7,  9, 10, 12, 11])},
{'max_depth': 3, 'min_child_weight': 1},
0.9933331724220043)
```

In [11]:
```
# min_child_weight and max_depth
param_test2 = {
 'max_depth':[4,5,6],
 'min_child_weight':[4,5,6]
}
gsearch2 = GridSearchCV(estimator = XGBClassifier( learning_rate=0.1, n_estimators=1484, max_depth=3,
 min_child_weight=1, gamma=0, subsample=0.8, colsample_bytree=0.8,
 objective= 'binary:logistic', nthread=4, scale_pos_weight=1,seed=27),
 param_grid = param_test2, scoring='roc_auc',n_jobs=4, cv=5)
gsearch2.fit(train[predictors],train[target])
gsearch2.cv_results_, gsearch2.best_params_, gsearch2.best_score_
```

```
C:\Users\admin\anaconda3\lib\site-packages\xgboost\sklearn.py:1224: UserWarning: The use of label encoder in X
GBClassifier is deprecated and will be removed in a future release. To remove this warning, do the following:
1) Pass option use_label_encoder=False when constructing XGBClassifier object; and 2) Encode your labels (y) a
s integers starting with 0, i.e. 0, 1, 2, ..., [num_class - 1].
  warnings.warn(label_encoder_deprecation_msg, UserWarning)

[13:40:23] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.5.1/src/learner.cc:1115: Starting
in XGBoost 1.3.0, the default evaluation metric used with the objective 'binary:logistic' was changed from 'er
ror' to 'logloss'. Explicitly set eval_metric if you'd like to restore the old behavior.
```

Out[11]:
```
({'mean_fit_time': array([ 952.62244406,  931.34406414,  993.40586224, 1255.31639724,
        1292.63907628, 1224.4488276 , 1550.41826186, 1476.49587364,
        1260.52944708]),
  'std_fit_time': array([ 13.40141049,  13.16698588,  44.72532571,  31.06582183,
         30.95756219,  12.04244689,  25.51923375,  21.44286864,
        322.663164  ]),
  'mean_score_time': array([1.00930686, 1.09055128, 1.20190148, 1.28014998, 3.77934427,
        1.25090842, 9.42779179, 1.79039021, 1.38934655]),
  'std_score_time': array([0.08534727, 0.19272686, 0.2683603 , 0.24837499, 2.07052474,
        0.05440438, 7.67327306, 0.39540293, 0.29376309]),
  'param_max_depth': masked_array(data=[4, 4, 4, 5, 5, 5, 6, 6, 6],
               mask=[False, False, False, False, False, False, False, False,
                     False],
         fill_value='?',
              dtype=object),
  'param_min_child_weight': masked_array(data=[4, 5, 6, 4, 5, 6, 4, 5, 6],
               mask=[False, False, False, False, False, False, False, False,
                     False],
         fill_value='?',
              dtype=object),
  'params': [{'max_depth': 4, 'min_child_weight': 4},
   {'max_depth': 4, 'min_child_weight': 5},
   {'max_depth': 4, 'min_child_weight': 6},
   {'max_depth': 5, 'min_child_weight': 4},
   {'max_depth': 5, 'min_child_weight': 5},
   {'max_depth': 5, 'min_child_weight': 6},
   {'max_depth': 6, 'min_child_weight': 4},
   {'max_depth': 6, 'min_child_weight': 5},
   {'max_depth': 6, 'min_child_weight': 6}],
  'split0_test_score': array([0.99527746, 0.99540918, 0.99526899, 0.99529308, 0.99541283,
        0.99549453, 0.99544469, 0.99577775, 0.99557315]),
  'split1_test_score': array([0.99858037, 0.99852784, 0.99858076, 0.99854772, 0.99855776,
        0.99855145, 0.99849639, 0.99858853, 0.99853959]),
  'split2_test_score': array([0.9992016 , 0.99923612, 0.9992491 , 0.99917695, 0.99920883,
        0.99919499, 0.99919234, 0.99920065, 0.99915955]),
  'split3_test_score': array([0.99214159, 0.99207326, 0.9921414 , 0.99196124, 0.9918384 ,
        0.99184588, 0.99149526, 0.9915091 , 0.99150372]),
  'split4_test_score': array([0.97956054, 0.98024202, 0.97937974, 0.97818197, 0.97851656,
        0.97821732, 0.97815327, 0.97785551, 0.97778082]),
  'mean_test_score': array([0.99295231, 0.99309768, 0.992924  , 0.99263219, 0.99270688,
        0.99266083, 0.99255639, 0.99258631, 0.99251137]),
  'std_test_score': array([0.00715726, 0.00691126, 0.00723276, 0.00767029, 0.0075624 ,
        0.00767729, 0.00769724, 0.00785007, 0.0078475 ]),
  'rank_test_score': array([2, 1, 3, 6, 4, 5, 8, 7, 9])},
 {'max_depth': 4, 'min_child_weight': 5},
 0.9930976819061339)
```

In [15]:
```
# min_child_weight
param_test2b = {'min_child_weight':[6,8,10,12]}
gsearch2b = GridSearchCV(estimator = XGBClassifier( learning_rate=0.1, n_estimators=1484, max_depth=4,
 min_child_weight=2, gamma=0, subsample=0.8, colsample_bytree=0.8,
 objective= 'binary:logistic', nthread=4, scale_pos_weight=1,seed=27),
 param_grid = param_test2b, scoring='roc_auc',n_jobs=4, cv=5)
gsearch2b.fit(train[predictors],train[target])
gsearch2b.cv_results_, gsearch2b.best_params_, gsearch2b.best_score_
```

```
C:\Users\admin\anaconda3\lib\site-packages\xgboost\sklearn.py:1224: UserWarning: The use of label encoder in X
GBClassifier is deprecated and will be removed in a future release. To remove this warning, do the following:
1) Pass option use_label_encoder=False when constructing XGBClassifier object; and 2) Encode your labels (y) a
s integers starting with 0, i.e. 0, 1, 2, ..., [num_class - 1].
  warnings.warn(label_encoder_deprecation_msg, UserWarning)

[20:03:43] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.5.1/src/learner.cc:1115: Starting
in XGBoost 1.3.0, the default evaluation metric used with the objective 'binary:logistic' was changed from 'er
ror' to 'logloss'. Explicitly set eval_metric if you'd like to restore the old behavior.
```

Out[15]:
```
({'mean_fit_time': array([1051.74756641,  946.71739564,  943.99094653,  925.68113279]),
  'std_fit_time': array([57.41024096, 18.3816269 , 13.36508591,  3.79335233]),
  'mean_score_time': array([6.00066304, 1.04559722, 0.95774961, 0.81208286]),
  'std_score_time': array([2.77842274, 0.03872624, 0.05874464, 0.14548765]),
  'param_min_child_weight': masked_array(data=[6, 8, 10, 12],
               mask=[False, False, False, False],
         fill_value='?',
              dtype=object),
  'params': [{'min_child_weight': 6},
   {'min_child_weight': 8},
   {'min_child_weight': 10},
   {'min_child_weight': 12}],
  'split0_test_score': array([0.99526899, 0.99521727, 0.99532411, 0.99543151]),
  'split1_test_score': array([0.99858076, 0.99852888, 0.99858611, 0.99858531]),
  'split2_test_score': array([0.9992491 , 0.9992114 , 0.99921756, 0.99920916]),
  'split3_test_score': array([0.9921414 , 0.99219999, 0.99207334, 0.9922621 ]),
  'split4_test_score': array([0.97937974, 0.97956487, 0.97956485, 0.97943044]),
  'mean_test_score': array([0.992924  , 0.99294448, 0.99295319, 0.9929837 ]),
  'std_test_score': array([0.00723276, 0.00714414, 0.00716401, 0.00721579]),
  'rank_test_score': array([4, 3, 2, 1])},
 {'min_child_weight': 12},
 0.9929837043972934)
```

In [16]:
```
# gamma
param_test3 = {
 'gamma':[i/10.0 for i in range(0,5)]
}
gsearch3 = GridSearchCV(estimator = XGBClassifier( learning_rate =0.1, n_estimators=1484, max_depth=4,
 min_child_weight=12, gamma=0, subsample=0.8, colsample_bytree=0.8,
 objective= 'binary:logistic', nthread=4, scale_pos_weight=1,seed=27),
 param_grid = param_test3, scoring='roc_auc',n_jobs=4, cv=5)
gsearch3.fit(train[predictors],train[target])
gsearch3.cv_results_, gsearch3.best_params_, gsearch3.best_score_
```

```
C:\Users\admin\anaconda3\lib\site-packages\xgboost\sklearn.py:1224: UserWarning: The use of label encoder in X
GBClassifier is deprecated and will be removed in a future release. To remove this warning, do the following:
1) Pass option use_label_encoder=False when constructing XGBClassifier object; and 2) Encode your labels (y) a
s integers starting with 0, i.e. 0, 1, 2, ..., [num_class - 1].
  warnings.warn(label_encoder_deprecation_msg, UserWarning)

[22:30:48] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.5.1/src/learner.cc:1115: Starting
in XGBoost 1.3.0, the default evaluation metric used with the objective 'binary:logistic' was changed from 'er
ror' to 'logloss'. Explicitly set eval_metric if you'd like to restore the old behavior.
```

Out[16]:
```
({'mean_fit_time': array([1352.5876718 , 1325.01373353, 1331.15502725, 1327.53921032,
        1137.33433867]),
  'std_fit_time': array([ 32.56213844,  12.6507868 ,  13.92604445,  13.03730149,
        362.38219525]),
  'mean_score_time': array([1.35761366, 1.39270434, 1.37685523, 1.37685852, 1.1583261 ]),
  'std_score_time': array([0.04465967, 0.02915961, 0.0438644 , 0.06813015, 0.1853099 ]),
  'param_gamma': masked_array(data=[0.0, 0.1, 0.2, 0.3, 0.4],
               mask=[False, False, False, False, False],
         fill_value='?',
              dtype=object),
  'params': [{'gamma': 0.0},
   {'gamma': 0.1},
   {'gamma': 0.2},
   {'gamma': 0.3},
   {'gamma': 0.4}],
  'split0_test_score': array([0.995556  , 0.99561346, 0.9954808 , 0.99545661, 0.99547685]),
  'split1_test_score': array([0.99855378, 0.99853726, 0.9985147 , 0.99856914, 0.99860776]),
  'split2_test_score': array([0.99913757, 0.99914831, 0.99920267, 0.99916078, 0.99917513]),
  'split3_test_score': array([0.99168043, 0.99157151, 0.99149873, 0.99156542, 0.99156905]),
  'split4_test_score': array([0.97842383, 0.97847199, 0.9780809 , 0.97785939, 0.97875852]),
  'mean_test_score': array([0.99267032, 0.99266851, 0.99255556, 0.99252227, 0.99271746]),
  'std_test_score': array([0.00759882, 0.00758745, 0.00773156, 0.00781231, 0.00748575]),
  'rank_test_score': array([2, 3, 4, 5, 1])},
 {'gamma': 0.4},
 0.992717460286984)
```

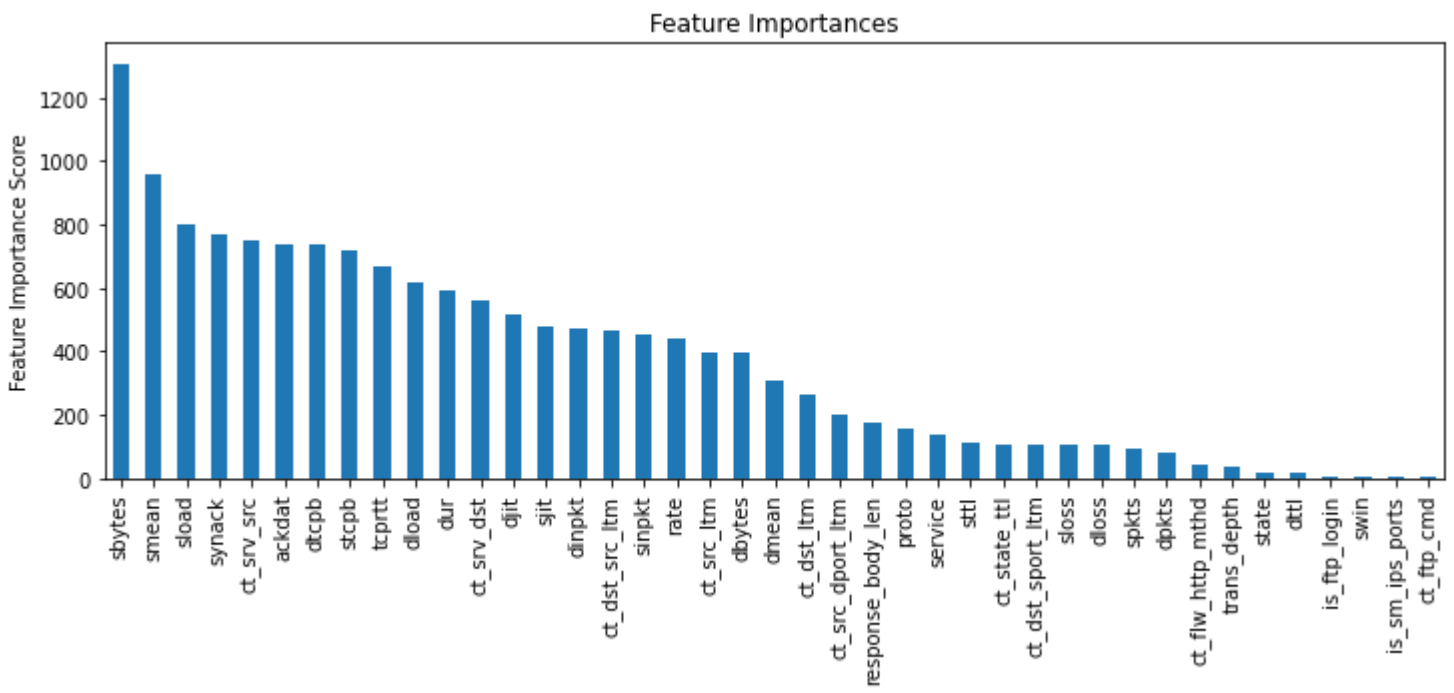
In [17]:
```
xgb2 = XGBClassifier(
        learning_rate =0.1,
        n_estimators=1484,
        max_depth=4,
        min_child_weight=12,
        gamma=0.4,
        subsample=0.8,
        colsample_bytree=0.8,
        objective= 'binary:logistic',
        nthread=4,
        scale_pos_weight=1,
        seed=27)
modelfit(xgb2, train, test, predictors)
```

```
      train-auc-mean  train-auc-std  test-auc-mean  test-auc-std
0           0.971057       0.000238       0.971007      0.001088
1           0.980660       0.000952       0.980480      0.001087
2           0.981464       0.000747       0.981225      0.000674
3           0.983026       0.000696       0.982838      0.001478
4           0.983694       0.000383       0.983567      0.000928
...              ...            ...            ...           ...
1479        0.997219       0.000108       0.993742      0.000335
1480        0.997222       0.000107       0.993742      0.000335
1481        0.997223       0.000107       0.993743      0.000334
1482        0.997224       0.000107       0.993743      0.000332
1483        0.997225       0.000107       0.993741      0.000333

[1484 rows x 4 columns]
1484
C:\Users\admin\anaconda3\lib\site-packages\xgboost\sklearn.py:1224: UserWarning: The use of label encoder in X
GBClassifier is deprecated and will be removed in a future release. To remove this warning, do the following:
1) Pass option use_label_encoder=False when constructing XGBClassifier object; and 2) Encode your labels (y) a
s integers starting with 0, i.e. 0, 1, 2, ..., [num_class - 1].
  warnings.warn(label_encoder_deprecation_msg, UserWarning)

[1 0 1 ... 1 1 1]
[0.5619081  0.02405899 0.73612547 ... 0.99961996 0.9999938  0.9999938 ]

Model Report
Accuracy : 0.9739
AUC Score (Train): 0.997224
AUC Score (Test): 0.982499
```



In [18]:
```
# subsample and colsample_bytree
param_test4 = {
 'subsample':[i/10.0 for i in range(6,10)],
 'colsample_bytree':[i/10.0 for i in range(6,10)]
}
gsearch4 = GridSearchCV(estimator = XGBClassifier( learning_rate =0.1, n_estimators=1484, max_depth=4,
 min_child_weight=12, gamma=0.4, subsample=0.8, colsample_bytree=0.8,
 objective= 'binary:logistic', nthread=4, scale_pos_weight=1,seed=27),
 param_grid = param_test4, scoring='roc_auc',n_jobs=4, cv=5)
gsearch4.fit(train[predictors],train[target])
gsearch4.cv_results_, gsearch4.best_params_, gsearch4.best_score_
```

```
C:\Users\admin\anaconda3\lib\site-packages\xgboost\sklearn.py:1224: UserWarning: The use of label encoder in X
GBClassifier is deprecated and will be removed in a future release. To remove this warning, do the following:
1) Pass option use_label_encoder=False when constructing XGBClassifier object; and 2) Encode your labels (y) a
s integers starting with 0, i.e. 0, 1, 2, ..., [num_class - 1].
  warnings.warn(label_encoder_deprecation_msg, UserWarning)

[04:10:47] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.5.1/src/learner.cc:1115: Starting
in XGBoost 1.3.0, the default evaluation metric used with the objective 'binary:logistic' was changed from 'er
ror' to 'logloss'. Explicitly set eval_metric if you'd like to restore the old behavior.
```

Out[18]:
```
({'mean_fit_time': array([ 748.54200873,  748.33966718,  754.1630981 ,  758.40649624,
          823.628093  ,  830.53890414,  836.6984724 ,  845.46243834,
          907.3901073 ,  914.08244901,  922.15662007,  931.83708663,
          989.34033957,  999.34366064, 1010.53597884, 1014.76679311]),
  'std_fit_time': array([ 1.4983344 ,  5.26715419,  7.09255571,  6.78998847,  2.66227534,
          5.41573563,  8.15318905,  5.12985675,  3.8220995 ,  7.64225501,
          6.78529294,  5.54525504,  5.22460208, 10.31855939, 11.81541678,
          5.15023143]),
  'mean_score_time': array([0.93164639, 0.88159971, 0.91281428, 1.06087508, 0.98185759,
         0.96294198, 0.90945134, 1.07278452, 0.92223482, 0.88476081,
         0.93682547, 0.95000062, 0.97560196, 1.05783224, 0.95388198,
         0.79203324]),
  'std_score_time': array([0.10161406, 0.09360645, 0.13115296, 0.14353376, 0.12975513,
         0.13066709, 0.07109229, 0.09577466, 0.05506742, 0.04910119,
         0.05081225, 0.04250329, 0.01816334, 0.10233687, 0.05231298,
         0.13575594]),
  'param_colsample_bytree': masked_array(data=[0.6, 0.6, 0.6, 0.6, 0.7, 0.7, 0.7, 0.7, 0.8, 0.8, 0.8,
                     0.8, 0.9, 0.9, 0.9, 0.9],
               mask=[False, False, False, False, False, False, False, False,
                     False, False, False, False, False, False, False, False],
         fill_value='?',
              dtype=object),
  'param_subsample': masked_array(data=[0.6, 0.7, 0.8, 0.9, 0.6, 0.7, 0.8, 0.9, 0.6, 0.7, 0.8,
                     0.9, 0.6, 0.7, 0.8, 0.9],
               mask=[False, False, False, False, False, False, False, False,
                     False, False, False, False, False, False, False, False],
         fill_value='?',
              dtype=object),
  'params': [{'colsample_bytree': 0.6, 'subsample': 0.6},
   {'colsample_bytree': 0.6, 'subsample': 0.7},
   {'colsample_bytree': 0.6, 'subsample': 0.8},
   {'colsample_bytree': 0.6, 'subsample': 0.9},
   {'colsample_bytree': 0.7, 'subsample': 0.6},
   {'colsample_bytree': 0.7, 'subsample': 0.7},
   {'colsample_bytree': 0.7, 'subsample': 0.8},
   {'colsample_bytree': 0.7, 'subsample': 0.9},
   {'colsample_bytree': 0.8, 'subsample': 0.6},
   {'colsample_bytree': 0.8, 'subsample': 0.7},
   {'colsample_bytree': 0.8, 'subsample': 0.8},
   {'colsample_bytree': 0.8, 'subsample': 0.9},
   {'colsample_bytree': 0.9, 'subsample': 0.6},
   {'colsample_bytree': 0.9, 'subsample': 0.7},
   {'colsample_bytree': 0.9, 'subsample': 0.8},
   {'colsample_bytree': 0.9, 'subsample': 0.9}],
  'split0_test_score': array([0.99554479, 0.99540444, 0.9953134 , 0.99540537, 0.99556765,
         0.99536201, 0.99533854, 0.99520396, 0.99546546, 0.99530061,
         0.99532459, 0.99523391, 0.99571331, 0.99530608, 0.99532394,
         0.99525396]),
  'split1_test_score': array([0.99859201, 0.99862751, 0.99850963, 0.99853401, 0.9985388 ,
         0.99853828, 0.99858273, 0.99855367, 0.99854511, 0.9986024 ,
         0.99858764, 0.99858337, 0.99847412, 0.99857433, 0.99854593,
         0.99851464]),
  'split2_test_score': array([0.99923542, 0.99921492, 0.99922552, 0.99924005, 0.99921583,
         0.99920275, 0.99920637, 0.99923386, 0.99925774, 0.99922761,
         0.99919391, 0.99922667, 0.99921915, 0.9992311 , 0.99920526,
         0.99922871]),
  'split3_test_score': array([0.99214907, 0.99210941, 0.99224552, 0.99227064, 0.99197351,
         0.99207958, 0.9922472 , 0.99230205, 0.99197197, 0.99211194,
         0.99230257, 0.99221693, 0.99189742, 0.99205655, 0.99214279,
         0.99216768]),
  'split4_test_score': array([0.98070632, 0.98086375, 0.98081285, 0.98021294, 0.98034917,
         0.9797488 , 0.98026854, 0.97973446, 0.97931407, 0.97977441,
         0.97962219, 0.980362  , 0.97994388, 0.97978843, 0.97966464,
         0.97936619]),
  'mean_test_score': array([0.99324552, 0.99324401, 0.99322138, 0.9931326 , 0.99312899,
         0.99298629, 0.99312868, 0.9930056 , 0.99291087, 0.99300339,
         0.99300618, 0.99312458, 0.99304958, 0.9929913 , 0.99297651,
         0.99290624]),
  'std_test_score': array([0.00675551, 0.00669118, 0.00668305, 0.00691813, 0.00688363,
```

```
        0.00708754, 0.00689534, 0.00708552, 0.0072706 , 0.00708756,
        0.00713361, 0.00685843, 0.00703806, 0.0070803 , 0.00711656,
        0.00722244]),
 'rank_test_score': array([ 1,  2,  3,  4,  5, 13,  6, 10, 15, 11,  9,  7,  8, 12, 14, 16])},
{'colsample_bytree': 0.6, 'subsample': 0.6},
0.9932455214231564)
```

In [21]:
```
# subsample and colsample_bytree
param_test5 = {
 'subsample':[i/100.0 for i in range(75,90,5)],
 'colsample_bytree':[i/100.0 for i in range(75,90,5)]
}
gsearch5 = GridSearchCV(estimator = XGBClassifier( learning_rate =0.1, n_estimators=1484, max_depth=4,
 min_child_weight=12, gamma=0.4, subsample=0.6, colsample_bytree=0.6,
 objective= 'binary:logistic', nthread=4, scale_pos_weight=1,seed=27),
 param_grid = param_test5, scoring='roc_auc',n_jobs=4,cv=5)
gsearch5.fit(train[predictors],train[target])
```

```
C:\Users\admin\anaconda3\lib\site-packages\xgboost\sklearn.py:1224: UserWarning: The use of label encoder in X
GBClassifier is deprecated and will be removed in a future release. To remove this warning, do the following:
1) Pass option use_label_encoder=False when constructing XGBClassifier object; and 2) Encode your labels (y) a
s integers starting with 0, i.e. 0, 1, 2, ..., [num_class - 1].
  warnings.warn(label_encoder_deprecation_msg, UserWarning)

[14:06:37] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.5.1/src/learner.cc:1115: Starting
in XGBoost 1.3.0, the default evaluation metric used with the objective 'binary:logistic' was changed from 'er
ror' to 'logloss'. Explicitly set eval_metric if you'd like to restore the old behavior.
```

Out[21]:
```
GridSearchCV(cv=5,
             estimator=XGBClassifier(base_score=None, booster=None,
                                     colsample_bylevel=None,
                                     colsample_bynode=None,
                                     colsample_bytree=0.6,
                                     enable_categorical=False, gamma=0.4,
                                     gpu_id=None, importance_type=None,
                                     interaction_constraints=None,
                                     learning_rate=0.1, max_delta_step=None,
                                     max_depth=4, min_child_weight=12,
                                     missing=nan, monotone_constraints=None,
                                     n_estimators=1484, n_jobs=None, nthread=4,
                                     num_parallel_tree=None, predictor=None,
                                     random_state=None, reg_alpha=None,
                                     reg_lambda=None, scale_pos_weight=1,
                                     seed=27, subsample=0.6, tree_method=None,
                                     validate_parameters=None, verbosity=None),
n_jobs=4,
param_grid={'colsample_bytree': [0.75, 0.8, 0.85],
            'subsample': [0.75, 0.8, 0.85]},
scoring='roc_auc')
```

In [22]:
```
# alpha (regularization paramter)
param_test7 = {
 'reg_alpha':[0, 0.001, 0.005, 0.01, 0.05]
}
gsearch7 = GridSearchCV(estimator = XGBClassifier( learning_rate =0.1, n_estimators=1484, max_depth=4,
 min_child_weight=12, gamma=0.4, subsample=0.6, colsample_bytree=0.6,
 objective= 'binary:logistic', nthread=4, scale_pos_weight=1,seed=27),
 param_grid = param_test7, scoring='roc_auc',n_jobs=4, cv=5)
gsearch7.fit(train[predictors],train[target])
gsearch7.cv_results_, gsearch7.best_params_, gsearch7.best_score_
```

```
C:\Users\admin\anaconda3\lib\site-packages\xgboost\sklearn.py:1224: UserWarning: The use of label encoder in X
GBClassifier is deprecated and will be removed in a future release. To remove this warning, do the following:
1) Pass option use_label_encoder=False when constructing XGBClassifier object; and 2) Encode your labels (y) a
s integers starting with 0, i.e. 0, 1, 2, ..., [num_class - 1].
  warnings.warn(label_encoder_deprecation_msg, UserWarning)

[15:42:43] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.5.1/src/learner.cc:1115: Starting
in XGBoost 1.3.0, the default evaluation metric used with the objective 'binary:logistic' was changed from 'er
ror' to 'logloss'. Explicitly set eval_metric if you'd like to restore the old behavior.
```

Out[22]:
```
({'mean_fit_time': array([812.81040015, 862.51195869, 811.7281651 , 780.75140615,
        645.04114671]),
  'std_fit_time': array([ 23.60141625,   5.64139992,  34.53603714,   2.12911998,
        204.03528189]),
  'mean_score_time': array([3.92816186, 3.65320826, 2.31476493, 1.01970897, 0.85922222]),
  'std_score_time': array([2.04482127, 2.32196406, 2.04028899, 0.07107676, 0.19822803]),
  'param_reg_alpha': masked_array(data=[0, 0.001, 0.005, 0.01, 0.05],
               mask=[False, False, False, False, False],
        fill_value='?',
             dtype=object),
  'params': [{'reg_alpha': 0},
   {'reg_alpha': 0.001},
   {'reg_alpha': 0.005},
   {'reg_alpha': 0.01},
   {'reg_alpha': 0.05}],
  'split0_test_score': array([0.99554479, 0.995498  , 0.99543291, 0.99561399, 0.99576716]),
  'split1_test_score': array([0.99859201, 0.99858212, 0.99857269, 0.99858684, 0.99858972]),
  'split2_test_score': array([0.99923542, 0.99923541, 0.99923938, 0.9992186 , 0.9991937 ]),
  'split3_test_score': array([0.99214907, 0.99226044, 0.99217492, 0.99211482, 0.99220565]),
  'split4_test_score': array([0.98070632, 0.98062124, 0.98022546, 0.98082633, 0.9806261 ]),
  'mean_test_score': array([0.99324552, 0.99323944, 0.99312907, 0.99327212, 0.99327647]),
  'std_test_score': array([0.00675551, 0.006779  , 0.00692373, 0.00671309, 0.00679156]),
  'rank_test_score': array([3, 4, 5, 2, 1])},
 {'reg_alpha': 0.05},
 0.9932764676762286)
```

In [ ]: