Robust Test - Features Groups - Forward, Backward

we will take each columns in out data and trasfer it with randomized vaules. the values will be in the range as the original column

Import

importing pandas package for handling data objects,

using pycaret only classification.

In [2]:
```
import pandas as pd
from pycaret.classification import *
import numpy as np
```

Functions and Constants

In [14]:
```
# set target feature
target_label = 'tuple'
# test imfuance over rf will satesfy
learning_models = ['rf']
# define numeric features which pycaret did not recognized
num_features = ['min_packet_size', 'min_fpkt', 'min_bpkt']
# set up features groups
SSL_features = ['fSSL_session_id_len', 'fSSL_num_extensions', 'fcipher_suites'
, 'ssl_v', ]
size_features = ['size_histogram_1','size_histogram_2','size_histogram_3',
                 'size_histogram_4','size_histogram_5','size_histogram_6',
                 'size_histogram_7','size_histogram_8','size_histogram_9', 'si
ze_histogram_10']
peak_features = ['fpeak_features_1','fpeak_features_2','fpeak_features_3',
                 'fpeak_features_4','fpeak_features_5','fpeak_features_6',
                 'fpeak_features_7','fpeak_features_8','fpeak_features_9',
                 'bpeak_features_1','bpeak_features_2','bpeak_features_3',
                 'bpeak_features_4','bpeak_features_5','bpeak_features_6',
                 'bpeak_features_7','bpeak_features_8','bpeak_features_9']
TCP_features = ['SYN_tcp_scale', 'SYN_tcp_winsize']
common_features = ['packet_count', 'fpackets', 'bpackets', 'fbytes', 'bbytes',
'num_keep_alive', 'mean_fttl']
stat_features = ['min_packet_size', 'max_packet_size', 'mean_packet_size',
                 'sizevar', 'std_fiat', # 'min_fiat', 'min_biat',
                'max_fiat','max_biat','std_biat','mean_fiat','mean_biat',
                'min_fpkt','min_bpkt','max_fpkt','max_bpkt','std_fpkt','std_bp
kt','mean_fpkt','mean_bpkt']
time_features = []
forward_features = ['fpeak_features_1','fpeak_features_2','fpeak_features_3',
'fpeak_features_4',
                    'fpeak_features_5','fpeak_features_6','fpeak_features_7',
'fpeak_features_8',
                    'fpeak_features_9','std_fiat','fpackets','fbytes','max_fia
t', #'min_fiat'
                    'mean_fiat','min_fpkt','max_fpkt','std_fpkt','mean_fpkt',
'fcipher_suites','ssl_v','mean_fttl']
backward_features = ['bpeak_features_1','bpeak_features_2','bpeak_features_3',
                    'bpeak_features_4','bpeak_features_5','bpeak_features_6',
                     'bpeak_features_7','bpeak_features_8','bpeak_features_9',
                     'bpackets','bbytes','max_biat','std_biat','mean_biat',
#'min_biat'
                     'min_bpkt','max_bpkt','std_bpkt']
both_features = ['fSSL_session_id_len','fSSL_num_extensions','SYN_tcp_scale',
                 'SYN_tcp_winsize','size_histogram_1','size_histogram_2',
                 'size_histogram_3','size_histogram_4','size_histogram_5',
                 'size_histogram_6','size_histogram_7','size_histogram_8',
                 'size_histogram_9','size_histogram_10','packet_count',
                 'min_packet_size','max_packet_size','mean_packet_size','sizev
ar','num_keep_alive']
```

Read Data

In [4]:
```
data = pd.read_csv(target_label+r'_dataset\new_all_features_'+target_label+'.csv',
                        sep='\t',
                        skiprows=[1])
```

Setup Classifier and Compare

In [4]:
```
setup(data=data,
      target=target_label,
      numeric_features=num_features,
      silent=True)
compare_models(whitelist=learning_models)
```

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
0	Random Forest Classifier	0.9719	0.0000	0.8644	0.9707	0.9703	0.9671	0.9671	4.8559

Out[4]:
```
OneVsRestClassifier(estimator=RandomForestClassifier(bootstrap=True,
                                                     ccp_alpha=0.0,
                                                     class_weight=None,
                                                     criterion='gini',
                                                     max_depth=None,
                                                     max_features='auto',
                                                     max_leaf_nodes=None,
                                                     max_samples=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=3050,
                                                     verbose=0,
                                                     warm_start=False),
                    n_jobs=-1)
```

take care of the Forward Features

In [18]:
```
# features_group = [x for x in forward_features if x not in peak_features]
features_group = forward_features
new_data = pd.DataFrame(columns=features_group+[target_label])
print ('current columns are : ' + str(features_group))
for i in features_group:
    new_data[i] = data[i]
```

```
current columns are : ['fpeak_features_1', 'fpeak_features_2', 'fpeak_feature
s_3', 'fpeak_features_4', 'fpeak_features_5', 'fpeak_features_6', 'fpeak_feat
ures_7', 'fpeak_features_8', 'fpeak_features_9', 'std_fiat', 'fpackets', 'fby
tes', 'max_fiat', 'mean_fiat', 'min_fpkt', 'max_fpkt', 'std_fpkt', 'mean_fpk
t', 'fcipher_suites', 'ssl_v', 'mean_fttl']
```

In [19]:
```
for i in features_group:
    max_value = 0
    value = 0
    min_value = new_data[i].values[0]
    for value in new_data[i]:
        if value > max_value: max_value = value
        if value < min_value: min_value = value
```

Setup and Check Only Forward Features

In [20]:
```
new_data[target_label] = data[target_label]
setup(data=new_data,
      target=target_label,
      silent=True)
compare_models(whitelist=learning_models)
```

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
0	Random Forest Classifier	0.9573	0.0000	0.8390	0.9570	0.9557	0.9501	0.9501	4.2610

Out[20]:
```
OneVsRestClassifier(estimator=RandomForestClassifier(bootstrap=True,
                                                     ccp_alpha=0.0,
                                                     class_weight=None,
                                                     criterion='gini',
                                                     max_depth=None,
                                                     max_features='auto',
                                                     max_leaf_nodes=None,
                                                     max_samples=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=6661,
                                                     verbose=0,
                                                     warm_start=False),
                    n_jobs=-1)
```

take care of the Backward Features

In [22]:
```
# features_group = [x for x in forward_features if x not in peak_features]
features_group = backward_features
new_data = pd.DataFrame(columns=features_group+[target_label])
print ('current columns are : ' + str(features_group))
for i in features_group:
    new_data[i] = data[i]
```

```
current columns are : ['bpeak_features_1', 'bpeak_features_2', 'bpeak_feature
s_3', 'bpeak_features_4', 'bpeak_features_5', 'bpeak_features_6', 'bpeak_feat
ures_7', 'bpeak_features_8', 'bpeak_features_9', 'bpackets', 'bbytes', 'max_b
iat', 'std_biat', 'mean_biat', 'min_bpkt', 'max_bpkt', 'std_bpkt']
```

In [23]:
```
for i in features_group:
    max_value = 0
    value = 0
    min_value = new_data[i].values[0]
    for value in new_data[i]:
        if value > max_value: max_value = value
        if value < min_value: min_value = value
```

Setup and Check Only Backward Features

In [24]:
```
new_data[target_label] = data[target_label]
setup(data=new_data,
      target=target_label,
      silent=True)
compare_models(whitelist=learning_models)
```

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
0	Random Forest Classifier	0.9465	0.0000	0.8004	0.9444	0.9437	0.9374	0.9375	1.2416

Out[24]:
```
OneVsRestClassifier(estimator=RandomForestClassifier(bootstrap=True,
                                                     ccp_alpha=0.0,
                                                     class_weight=None,
                                                     criterion='gini',
                                                     max_depth=None,
                                                     max_features='auto',
                                                     max_leaf_nodes=None,
                                                     max_samples=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=4996,
                                                     verbose=0,
                                                     warm_start=False),
                    n_jobs=-1)
```

take care of the Forward+BOTH Features

In [25]:
```
# features_group = [x for x in forward_features if x not in peak_features]
features_group = forward_features + both_features
new_data = pd.DataFrame(columns=features_group+[target_label])
print ('current columns are : ' + str(features_group))
for i in features_group:
    new_data[i] = data[i]
```

```
current columns are : ['fpeak_features_1', 'fpeak_features_2', 'fpeak_feature
s_3', 'fpeak_features_4', 'fpeak_features_5', 'fpeak_features_6', 'fpeak_feat
ures_7', 'fpeak_features_8', 'fpeak_features_9', 'std_fiat', 'fpackets', 'fby
tes', 'max_fiat', 'mean_fiat', 'min_fpkt', 'max_fpkt', 'std_fpkt', 'mean_fpk
t', 'fcipher_suites', 'ssl_v', 'mean_fttl', 'fSSL_session_id_len', 'fSSL_num_
extensions', 'SYN_tcp_scale', 'SYN_tcp_winsize', 'size_histogram_1', 'size_hi
stogram_2', 'size_histogram_3', 'size_histogram_4', 'size_histogram_5', 'size
_histogram_6', 'size_histogram_7', 'size_histogram_8', 'size_histogram_9', 's
ize_histogram_10', 'packet_count', 'min_packet_size', 'max_packet_size', 'mea
n_packet_size', 'sizevar', 'num_keep_alive']
```

In [26]:
```
for i in features_group:
    max_value = 0
    value = 0
    min_value = new_data[i].values[0]
    for value in new_data[i]:
        if value > max_value: max_value = value
        if value < min_value: min_value = value
```

Setup and Check Only Forward+BOTH Features

In [27]:
```
new_data[target_label] = data[target_label]
setup(data=new_data,
      target=target_label,
      silent=True)
compare_models(whitelist=learning_models)
```

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
0	Random Forest Classifier	0.9723	0.0000	0.8712	0.9716	0.9710	0.9676	0.9676	4.2966

Out[27]:
```
OneVsRestClassifier(estimator=RandomForestClassifier(bootstrap=True,
                                                     ccp_alpha=0.0,
                                                     class_weight=None,
                                                     criterion='gini',
                                                     max_depth=None,
                                                     max_features='auto',
                                                     max_leaf_nodes=None,
                                                     max_samples=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=3515,
                                                     verbose=0,
                                                     warm_start=False),
                    n_jobs=-1)
```

take care of the Backward+BOTH Features

In [28]:
```
# features_group = [x for x in forward_features if x not in peak_features]
features_group = backward_features + both_features
new_data = pd.DataFrame(columns=features_group+[target_label])
print ('current columns are : ' + str(features_group))
for i in features_group:
    new_data[i] = data[i]
```

```
current columns are : ['bpeak_features_1', 'bpeak_features_2', 'bpeak_feature
s_3', 'bpeak_features_4', 'bpeak_features_5', 'bpeak_features_6', 'bpeak_feat
ures_7', 'bpeak_features_8', 'bpeak_features_9', 'bpackets', 'bbytes', 'max_b
iat', 'std_biat', 'mean_biat', 'min_bpkt', 'max_bpkt', 'std_bpkt', 'fSSL_sess
ion_id_len', 'fSSL_num_extensions', 'SYN_tcp_scale', 'SYN_tcp_winsize', 'size
_histogram_1', 'size_histogram_2', 'size_histogram_3', 'size_histogram_4', 's
ize_histogram_5', 'size_histogram_6', 'size_histogram_7', 'size_histogram_8',
'size_histogram_9', 'size_histogram_10', 'packet_count', 'min_packet_size',
'max_packet_size', 'mean_packet_size', 'sizevar', 'num_keep_alive']
```

In [29]:
```
for i in features_group:
    max_value = 0
    value = 0
    min_value = new_data[i].values[0]
    for value in new_data[i]:
        if value > max_value: max_value = value
        if value < min_value: min_value = value
```

Setup and Check Only Backward+BOTH Features

In [30]:
```
new_data[target_label] = data[target_label]
setup(data=new_data,
      target=target_label,
      silent=True)
compare_models(whitelist=learning_models)
```

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
0	Random Forest Classifier	0.9696	0.0000	0.8661	0.9692	0.9683	0.9644	0.9645	4.3103

Out[30]:
```
OneVsRestClassifier(estimator=RandomForestClassifier(bootstrap=True,
                                                     ccp_alpha=0.0,
                                                     class_weight=None,
                                                     criterion='gini',
                                                     max_depth=None,
                                                     max_features='auto',
                                                     max_leaf_nodes=None,
                                                     max_samples=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=1256,
                                                     verbose=0,
                                                     warm_start=False),
                    n_jobs=-1)
```

take care of the Forward Without peaks Features

In [31]:
```
features_group = [x for x in forward_features if x not in peak_features]
new_data = pd.DataFrame(columns=features_group+[target_label])
print ('current columns are : ' + str(features_group))
for i in features_group:
    new_data[i] = data[i]
```

```
current columns are : ['std_fiat', 'fpackets', 'fbytes', 'max_fiat', 'mean_fi
at', 'min_fpkt', 'max_fpkt', 'std_fpkt', 'mean_fpkt', 'fcipher_suites', 'ssl_
v', 'mean_fttl']
```

In [32]:
```
for i in features_group:
    max_value = 0
    value = 0
    min_value = new_data[i].values[0]
    for value in new_data[i]:
        if value > max_value: max_value = value
        if value < min_value: min_value = value
```

Setup and Check Forward Without peaks Features

In [33]:
```
new_data[target_label] = data[target_label]
setup(data=new_data,
      target=target_label,
      silent=True)
compare_models(whitelist=learning_models)
```

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
0	Random Forest Classifier	0.9554	0.0000	0.8278	0.9548	0.9537	0.9478	0.9479	1.0616

Out[33]:
```
OneVsRestClassifier(estimator=RandomForestClassifier(bootstrap=True,
                                                     ccp_alpha=0.0,
                                                     class_weight=None,
                                                     criterion='gini',
                                                     max_depth=None,
                                                     max_features='auto',
                                                     max_leaf_nodes=None,
                                                     max_samples=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=8113,
                                                     verbose=0,
                                                     warm_start=False),
                    n_jobs=-1)
```

take care of the Forward+COMMON+STAT+BOTH without Peaks Features

In [34]:
```
forward_stat_features = [x for x in stat_features if x in forward_features]
forward_common_features = [x for x in common_features if x in forward_features
]
features_group = forward_stat_features + forward_common_features + both_featur
es
features_group = [x for x in features_group if x not in peak_features]
new_data = pd.DataFrame(columns=features_group+[target_label])
print ('current columns are : ' + str(features_group))
for i in features_group:
    new_data[i] = data[i]
```

```
current columns are : ['std_fiat', 'max_fiat', 'mean_fiat', 'min_fpkt', 'max_
fpkt', 'std_fpkt', 'mean_fpkt', 'fpackets', 'fbytes', 'mean_fttl', 'fSSL_sess
ion_id_len', 'fSSL_num_extensions', 'SYN_tcp_scale', 'SYN_tcp_winsize', 'size
_histogram_1', 'size_histogram_2', 'size_histogram_3', 'size_histogram_4', 's
ize_histogram_5', 'size_histogram_6', 'size_histogram_7', 'size_histogram_8',
'size_histogram_9', 'size_histogram_10', 'packet_count', 'min_packet_size',
'max_packet_size', 'mean_packet_size', 'sizevar', 'num_keep_alive']
```

In [35]:
```
for i in features_group:
    max_value = 0
    value = 0
    min_value = new_data[i].values[0]
    for value in new_data[i]:
        if value > max_value: max_value = value
        if value < min_value: min_value = value
```

Setup and Check Forward+COMMON+STAT+BOTH without Peaks Features

In [36]:
```
new_data[target_label] = data[target_label]
setup(data=new_data,
      target=target_label,
      silent=True)
compare_models(whitelist=learning_models)
```

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
0	Random Forest Classifier	0.9726	0.0000	0.8733	0.9720	0.9713	0.9679	0.9679	4.2348

Out[36]:
```
OneVsRestClassifier(estimator=RandomForestClassifier(bootstrap=True,
                                                     ccp_alpha=0.0,
                                                     class_weight=None,
                                                     criterion='gini',
                                                     max_depth=None,
                                                     max_features='auto',
                                                     max_leaf_nodes=None,
                                                     max_samples=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=4391,
                                                     verbose=0,
                                                     warm_start=False),
                    n_jobs=-1)
```

so this is the mininal feature set we can use, note the size hist can be shrinked into 3 columns

In [ ]:
```

```