Features Groups - Inverted

Import

In [1]:

```
import pandas as pd
from pycaret.classification import *
import numpy as np
```

Functions and Constants

In [2]:

```
# 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','mean_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 [3]: data = pd.read_csv(target_label+r'_dataset\new_all_features_'+target_label+'.c
        sv',
                               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	0.9724	0.0000	0.8674	0.9721	0.9713	0.9677	0.9678	5.1242

```
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=2092,
                                                             verbose=0,
                                                             warm_start=False),
                            n_jobs=-1)
```

Create new column for any features

In [5]:

```
invert_features_len = len(forward_features)
for i in range(0,invert_features_len-3):
    data[forward_features[i]] = data[forward_features[i]] - data[backward_feat
ures[i]]
for i in peak_features:
    data = data.drop(i, axis=1)
for i in backward_features:
    if i in data.columns:
        data = data.drop(i, axis=1)
```

In [6]:

```
setup(data=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	0.9693	0.0000	0.8626	0.9688	0.9678	0.9641	0.9641	4.5093

Out[6]:

```
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=1066,
                                                     verbose=0,
                                                     warm_start=False),
                    n_jobs=-1)
```

In [ ]: