2.2 Backwards Feature Selection:

Step backwards feature selection, as the name suggests is the exact opposite of step forward feature selection that we studied in the last section. In the first step of the step backwards feature selection, one feature is removed in round-robin fashion from the feature set and the performance of the classifier is evaluated.

In [5]:

```
# removing correlation variables
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
```

In [6]:

```
# read the data
data = pd.read_csv('paribas.csv',nrows= 20000)
```

In [12]:

```
num_cols = ['int16','int32','int64','float16','float32','float64']
numerical_cols = list(data.select_dtypes(include = num_cols).columns)
data =data[numerical_cols]
```

In [13]:

```
train_features,test_features,train_labels,test_labels = train_test_split(
data.drop(labels = ['target','ID'] ,axis=1),
data['target'],
test_size = 0.2,
random_state = 41)
```

In [15]:

```
correlated_features=set()
correlation_matrix = data.corr()
```

In [18]:

```
for i in range(len(correlation_matrix.columns)):
    for j in range(i):
        if abs(correlation_matrix.iloc[i,j]) > 0.8:
            col_name = correlation_matrix.columns[i]
            correlated_features.add(col_name)
```

In [19]:

```
train_features.drop(labels = correlated_features , axis =1, inplace = True)
test_features.drop(labels = correlated_features , axis =1 , inplace = True)
```

Backward Feature Selection

In [26]:

```
from sklearn.metrics import roc_auc_score
from mlxtend.feature_selection import SequentialFeatureSelector as sfs
from sklearn.ensemble import RandomForestClassifier as rfc
```

The only change here is we make forward = false for backward_feature_selection

In [28]:

```
feature_selector = sfs(rfc(n_jobs = 1),
                      k_features = 15,
                      forward = False,
                      verbose = 2,
                      floating = False,
                      scoring ='roc_auc',
                      cv = 4)
```

In [29]:

```
features = feature_selector.fit(train_features.fillna(0),train_labels)
```

```
\sklearn\ensemble\forest.py:245: FutureWarning: The default value of n_
estimators will change from 10 in version 0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
c:\users\dell\appdata\local\programs\python\python36\lib\site-packages

\sklearn\ensemble\forest.py:245: FutureWarning: The default value of n_
estimators will change from 10 in version 0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
c:\users\dell\appdata\local\programs\python\python36\lib\site-packages
\sklearn\ensemble\forest.py:245: FutureWarning: The default value of n_
estimators will change from 10 in version 0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
c:\users\dell\appdata\local\programs\python\python36\lib\site-packages
\sklearn\ensemble\forest.py:245: FutureWarning: The default value of n_
estimators will change from 10 in version 0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
c:\users\dell\appdata\local\programs\python\python36\lib\site-packages
\sklearn\ensemble\forest.py:245: FutureWarning: The default value of n_
estimators will change from 10 in version 0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
```

In [34]:

```
filtered2_features = train_features.columns[list(features.k_feature_idx_)]
```

In [43]:

```
clf = RandomForestClassifier(n_estimators = 100 , random_state = 41 , max_depth =3)
clf.fit(train_features[filtered2_features].fillna(0),train_labels)

train_pred = clf.predict_proba(train_features[filtered2_features].fillna(0))
print('Accuracy on training_data:{}'.format(roc_auc_score(train_labels , train_pred[:,1

test_pred = clf.predict_proba(test_features[filtered2_features].fillna(0))
print('Test Accuracy:{}'.format(roc_auc_score(test_labels , test_pred[:,1])))
```

```
Accuracy on training_data:0.7079639398257904
Test Accuracy:0.7124272648835203
```

links for the methods() used

SequentialFeatureSelector(): https://rasbt.github.io/mlxtend/user_guide/feature_selection/SequentialFeatureSelector/#sequential-feature-selector (https://rasbt.github.io/mlxtend/user_guide/feature_selection/SequentialFeatureSelector/#sequential-feature-selector)

RandomForestClassifier(): https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html (https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html)