# 特征选择算法实验报告

实验目的：（根据模型的类别来选择最佳变量数目和最佳变量，暂无王老师文档）

算法API及测试结果：

Feature Selection Tool

@date:2017-08-28

@author:charleshen

**File:**

feature\_selection.py

**main function:**

SFE(df, estimator, param\_grid, im\_method = 'seq', sel\_method = 'sfe',

vip\_feat = 20, Forward = True, max\_feat = 50, batch\_feat = 1,

para\_search\_step = 1, random\_state = None, n\_jobs = 4,

cv = 5, randcv = True)

**PARAS1**: parameter of im\_method, get feature importance of each feature, support:

'fscore': F-score of features

'pcc': Person corr. Coefficient of features

'tree': optimal tree’s feature importance(gini importance)

'lasso': weights of features for optimal lasso regression model

'elnet': weights of features for optimal elnet regression model

'seq': sequence of the features(default, namely feature importance is feature’s sequence(order) )

**PARAS2:**  parameter of sel\_method, method of get best feature numbers, support:

'sfe': Stepwise Feature Elimination, if Forward is True, then feature addiation(add features one by one using recursive try algorithm)

'ofe': Ordered Feature Elimination, if Forward is True, then feature addiation(add features one by one with the original order or feature importance order)

**PARAS3:** other parameters:

vip\_feat = 20: very important features, will be fixed in the elimination or addition

Forward = True: if forward is true, features will be added one by one instead of elimination

max\_feat = 50: max number of features

batch\_feat = 1: batch size of features for addition or elimination

para\_search\_step = 1: step of search best parameters during addition or elimination

random\_state = None

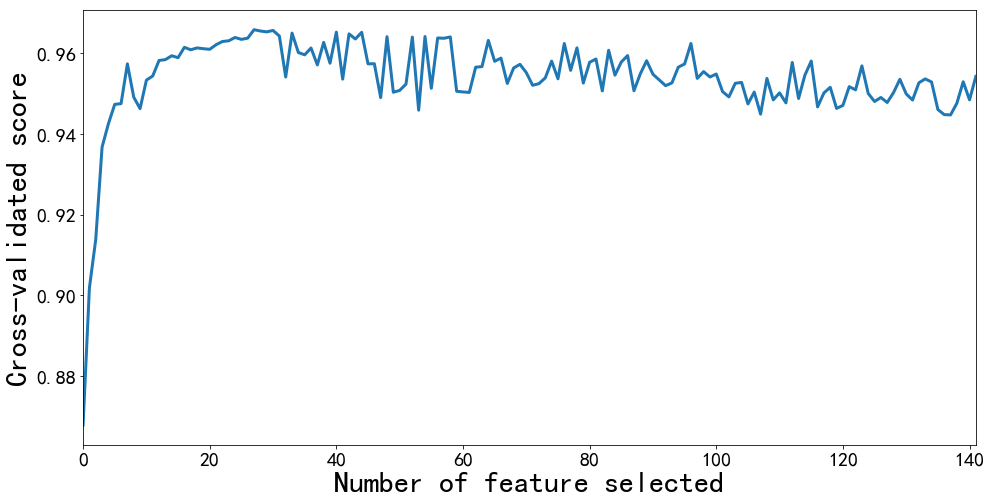
n\_jobs = 4: number of threads for parallelization

cv = 5: cross validation’s fold, default cross validated score is ROC-AUC for classification, R-squared for regression

randcv = True: if True, then random search method will be used in the grid-search.

**Test:**

the tool has been tested on dataset of bairong(bairong\_train.csv)

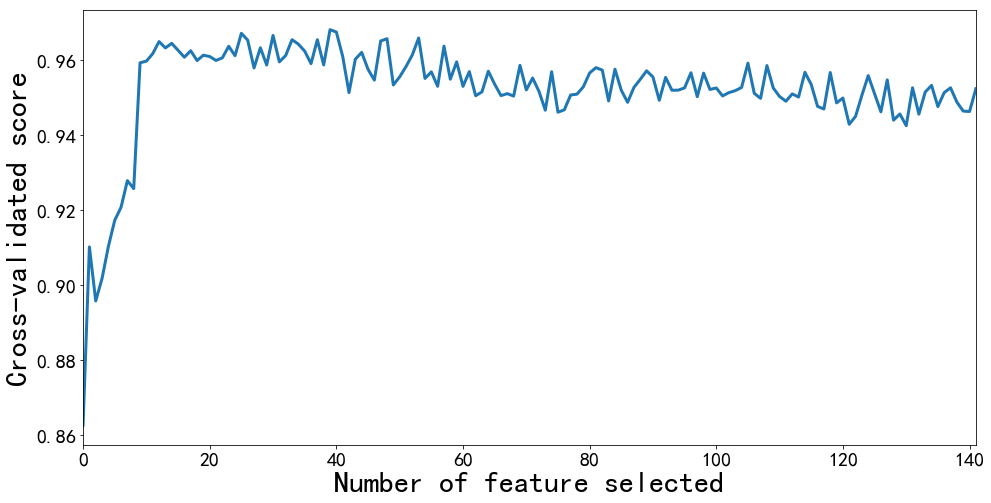


**OFE test:**

**estimator = SVM()**

**im\_method= ‘tree’**

**sel\_method = ‘ofe’:**

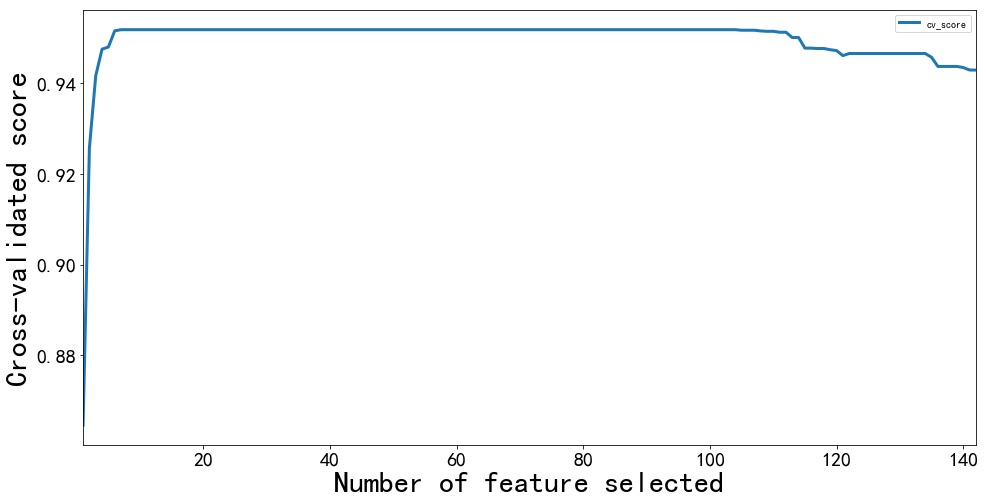


**estimator = SVM() im\_method= ‘lasso’**

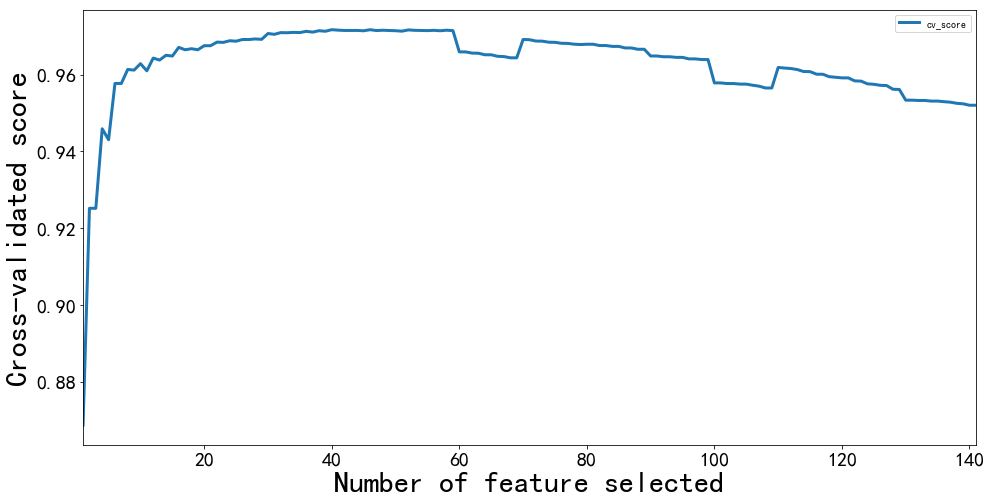
**sel\_method = ‘ofe’**

**SFE test:**

**estimator = DecisionTreeClassifier()**

 **im\_method= ‘tree’**

**sel\_method = ‘sfe’**

****

**estimator = SVM()**

**im\_method= ‘tree’**

**sel\_method = ‘sfe’**

**Conclusion:**

Both OFE and SFE can be used to select best number of features to avoid over- fitting;

SFE is a better method to select best number of features(Because SFE can raise cross validated score step by step)

OFE is more faster than SFE