In [46]:

```
from sklearn.datasets import make_friedman1,fetch_california_housing
from sklearn.feature_selection import RFE
from sklearn.linear_model import LinearRegression
estimator = LinearRegression()
```

In [47]:

```
X, y = make_friedman1(n_samples=50, n_features=10, random_state=0)
```

RFE

In [48]:

```
X.shape,y.shape
```

Out[48]:

```
((50, 10), (50,))
```

In [49]:

```
selector = RFE(estimator, n_features_to_select=3, step=1)
selector = selector.fit(X,y)
```

In [50]:

```
selector.ranking_
```

Out[50]:

```
array([3, 2, 1, 1, 1, 4, 5, 8, 7, 6])
```

In [51]:

```
X = selector.transform(X)
X.shape
```

Out[51]:

```
(50, 3)
```

SelectFromModal

In [52]:

```
from sklearn.feature_selection import SelectFromModel
```

In [53]:

```
dataset = fetch_california_housing()
estimator = LinearRegression()
```

In [54]:

```
X = dataset.data
y = dataset.target
```

In [55]:

```
X.shape
```

Out[55]:

```
(20640, 8)
```

In [56]:

```
selector = SelectFromModel(estimator).fit(X, y)
print("Threshold: ",selector.threshold_)
```

```
Threshold:  0.25726693216557395
```

In [57]:

```
X = selector.transform(X)
X.shape
```

Out[57]:

```
(20640, 4)
```

SequentialFeatureSelection

```
In [58]: from sklearn.feature_selection import SequentialFeatureSelector
```

```
In [59]: sfs = SequentialFeatureSelector(estimator, n_features_to_select=3)
```

```
In [60]: sfs.fit(X, y)
         sfs.transform(X).shape
```

```
Out[60]: (20640, 3)
```