In [1]:
```
from sklearn.datasets import load_iris
```

In [2]:
```
iris = load_iris()
```

In [3]:
```
iris.keys()
```

Out[3]:
```
dict_keys(['data', 'target', 'target_names', 'DESCR', 'feature_names', 'filen
ame'])
```

In [4]:
```
iris['feature_names']
```

Out[4]:
```
['sepal length (cm)',
 'sepal width (cm)',
 'petal length (cm)',
 'petal width (cm)']
```

In [5]:
```
X =  iris['data']
y = iris['target']
```

In [6]:
```
X.shape
```

Out[6]:
```
(150, 4)
```

In [7]:
```
y.shape
```

Out[7]:
```
(150,)
```

In [8]:
```
from sklearn.feature_selection import SelectKBest
```

compute the ANOVA F-value between each feature and the target vector.

The F-value scores examine if, when we group the numerical feature by the target vector, the means for each group are significantly different.

In [9]:
```
from sklearn.feature_selection import f_classif
# Create an SelectKBest object to select features with two best ANOVA F-Values
X_new_F = SelectKBest(f_classif, k=2).fit_transform(X,y)
X_new_F.shape
```

Out[9]:
```
(150, 2)
```

In [ ]:

In [ ]:

In [10]:
```
from sklearn.datasets import load_digits

X, y = load_digits(return_X_y=True)
```

In [11]:
```
X.shape
```

Out[11]:
```
(1797, 64)
```

In [12]:
```
y.shape
```

Out[12]:
```
(1797,)
```

In [ ]:
```

```

## chi2

calculate chi2 between each feature and target value

(categorical features)

In [13]:
```
from sklearn.feature_selection import chi2
## Select 20 features with highest chi-squared statistics
X_new_chi2 = SelectKBest(chi2, k=20).fit_transform(X,y)
X_new_chi2.shape
```

Out[13]:
```
(1797, 20)
```

In [ ]:
```

```

## Mutual Information

Mutual Information between two variables measures the dependence of one variable to another.

- If X and Y are two variables, and If X and Y are independent, then no information about Y can be obtained by knowing X or vice versa. Hence their mutual information is 0.
- X is a deterministic function of Y, then we can determine X from Y and Y from X with mutual information 1.
- we have Y = f(X,Z,M,N), 0 < mutual information <

We can select our features from feature space by ranking their mutual information with the target variable.

F-Test captures the linear relationship well. Mutual Information captures any kind of relationship between two variables

In [15]:
```
from sklearn.feature_selection import mutual_info_classif
X_new_mut = SelectKBest(mutual_info_classif, k=15).fit_transform(X,y)
X_new_mut.shape
```

Out[15]:
```
(1797, 15)
```