In [5]:
```
import pandas as pd
import numpy as np
from sklearn.datasets import load_iris
from sklearn.model_selection import cross_val_score
```

In [7]:
```
iris = load_iris()
```

In [11]:
```
iris.feature_names
```

Out[11]:
```
['sepal length (cm)',
 'sepal width (cm)',
 'petal length (cm)',
 'petal width (cm)']
```

In [12]:
```
iris.data
```

```
       [6.9, 3.2, 5.7, 2.3],
       [5.6, 2.8, 4.9, 2. ],
       [7.7, 2.8, 6.7, 2. ],
       [6.3, 2.7, 4.9, 1.8],
       [6.7, 3.3, 5.7, 2.1],
       [7.2, 3.2, 6. , 1.8],
       [6.2, 2.8, 4.8, 1.8],
       [6.1, 3. , 4.9, 1.8],
       [6.4, 2.8, 5.6, 2.1],
       [7.2, 3. , 5.8, 1.6],
       [7.4, 2.8, 6.1, 1.9],
       [7.9, 3.8, 6.4, 2. ],
       [6.4, 2.8, 5.6, 2.2],
       [6.3, 2.8, 5.1, 1.5],
       [6.1, 2.6, 5.6, 1.4],
       [7.7, 3. , 6.1, 2.3],
       [6.3, 3.4, 5.6, 2.4],
       [6.4, 3.1, 5.5, 1.8],
       [6. , 3. , 4.8, 1.8],
       [6.9, 3.1, 5.4, 2.1],
```

In [13]:
```
from sklearn.linear_model import LogisticRegression
```

In [14]:
```
lr_scores = cross_val_score(LogisticRegression(),iris.data,iris.target,cv=4
lr_scores
```

```
C:\Users\nithi\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.
py:762: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown i
n:
    https://scikit-learn.org/stable/modules/preprocessing.html (https://sc
ikit-learn.org/stable/modules/preprocessing.html)
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-reg
ression (https://scikit-learn.org/stable/modules/linear_model.html#logisti
c-regression)
  n_iter_i = _check_optimize_result(
```

Out[14]:
```
array([0.97368421, 0.97368421, 0.94594595, 1.        ])
```

```
In [15]: from sklearn.svm import SVC
```

```
In [16]: svm_scores = cross_val_score(SVC(),iris.data,iris.target,cv=4)
         svm_scores
```

```
Out[16]: array([0.97368421, 1.        , 0.94594595, 0.97297297])
```

```
In [17]: from sklearn.tree import DecisionTreeClassifier
```

```
In [18]: DC_Score = cross_val_score(DecisionTreeClassifier(),iris.data,iris.target,c
         DC_Score
```

```
Out[18]: array([0.97368421, 0.94736842, 0.94594595, 0.97297297])
```

```
In [19]: from sklearn.ensemble import RandomForestClassifier
```

```
In [20]: RandomF_Score = cross_val_score(RandomForestClassifier(),iris.data,iris.tar
         RandomF_Score
```

```
Out[20]: array([0.97368421, 0.94736842, 0.94594595, 1.        ])
```

```
In [21]: np.average(lr_scores)
```

```
Out[21]: 0.9733285917496444
```

```
In [22]: np.average(svm_scores)
```

```
Out[22]: 0.9731507823613088
```

```
In [23]: np.average(DC_Score)
```

```
Out[23]: 0.9599928876244666
```

```
In [24]: np.average(RandomF_Score)
```

```
Out[24]: 0.9667496443812233
```