In [1]:

```
import numpy as np
import pandas as pd
```

In [3]:

```
master_iris = pd.read_csv(r"D:\PG-DAI\MachineLearning\Dec 27 Boosting\iris.csv")
```

In [4]:

```
master_iris.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 149 entries, 0 to 148
Data columns (total 5 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   5.1          149 non-null    float64
 1   3.5          149 non-null    float64
 2   1.4          149 non-null    float64
 3   0.2          149 non-null    float64
 4   Iris-setosa  149 non-null    object 
dtypes: float64(4), object(1)
memory usage: 5.9+ KB
```

In [10]:

```
master_iris.iloc[:,:-1]
```

Out[10]:

	5.1	3.5	1.4	0.2
0	4.9	3.0	1.4	0.2
1	4.7	3.2	1.3	0.2
2	4.6	3.1	1.5	0.2
3	5.0	3.6	1.4	0.2
4	5.4	3.9	1.7	0.4
...	...	...	...	...
144	6.7	3.0	5.2	2.3
145	6.3	2.5	5.0	1.9

	5.1	3.5	1.4	0.2
146	6.5	3.0	5.2	2.0
147	6.2	3.4	5.4	2.3
148	5.9	3.0	5.1	1.8

149 rows × 4 columns

In [11]:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(master_iris.iloc[:,:-1],master_iris.iloc[:,-1], random_state=7)
```

In [12]:

```
from sklearn.ensemble import BaggingClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import cross_val_score
```

In [14]:

```
dtc = DecisionTreeClassifier()
model = BaggingClassifier(base_estimator=dtc,n_estimators=100)
results = cross_val_score(model,master_iris.iloc[:,:-1],master_iris.iloc[:,-1], cv = 10)
print(results.mean())
```

```
0.9666666666666666
```

In [15]:

```
from sklearn.ensemble import AdaBoostClassifier

model = AdaBoostClassifier(n_estimators=100, random_state=42)
results = cross_val_score(model,master_iris.iloc[:,:-1],master_iris.iloc[:,-1], cv = 10)

print(results.mean())
```

```
0.9466666666666667
```

In [16]:

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.svm import SVC
from sklearn.ensemble import VotingClassifier
from sklearn.naive_bayes import GaussianNB
```

In [18]:

```
# create the sub models
estimators = []
model1 = GaussianNB()
estimators.append(('Naive_Bais', model1))
model2 = DecisionTreeClassifier()
estimators.append(('cart', model2))
model3 = SVC()
estimators.append(('svm', model3))
# create the ensemble model
ensemble = VotingClassifier(estimators)
results = cross_val_score(ensemble,master_iris.iloc[:,:-1],master_iris.iloc[:,-1], cv= 10)
print(results.mean())
```

```
0.9666666666666666
```

In [21]:

```
from sklearn.ensemble import GradientBoostingClassifier
model = GradientBoostingClassifier()
results = cross_val_score(model, master_iris.iloc[:,:-1],master_iris.iloc[:,-1], cv = 10)
print(results.mean())
```

```
0.96
```