In [ ]:
```
Chapter 6 - Other Popular Machine Learning Methods

Part 5 - Ensemble methods with random forest

This is classification problem, where in we weill be estimating the species label
```

In [2]:
```
import numpy as np
import pandas as pd

import sklearn.datasets as datasets
from sklearn.model_selection import train_test_split
from sklearn import metrics
```

In [4]:
```
from sklearn.ensemble import RandomForestClassifier
```

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

df = pd.DataFrame(iris.data, columns=iris.feature_names)
y = pd.DataFrame(iris.target)

y.columns = ['labels']

print(df.head())
y[0:5]
```

```
   sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)
0                5.1               3.5                1.4               0.2
1                4.9               3.0                1.4               0.2
2                4.7               3.2                1.3               0.2
3                4.6               3.1                1.5               0.2
4                5.0               3.6                1.4               0.2
```

Out[7]:

	labels
0	0
1	0
2	0
3	0
4	0

```
The data set contains information on the:

sepal length (cm)
sepal width (cm)
petal length (cm)
petal width (cm)
species type
```

In [8]:
```
df.isnull().any()==True
```

Out[8]:
```
sepal length (cm)    False
sepal width (cm)     False
petal length (cm)    False
petal width (cm)     False
dtype: bool
```

In [9]:
```
print(y.labels.value_counts())
```

```
2    50
1    50
0    50
Name: labels, dtype: int64
```

```
Preparing the data for training the model
```

In [12]:
```
x_train, x_test, y_train, y_test = train_test_split(df,y,test_size=.2, random_sta
```

```
Build a Random Forest Model
```

In [13]:
```
classifier = RandomForestClassifier(n_estimators=200, random_state=0)

y_train_array = np.ravel(y_train)

classifier.fit(x_train, y_train_array)

y_pred = classifier.predict(x_test)
```

```
Evaluating the model on the test data
```

In [14]:
```
print(metrics.classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00         7
           1       0.92      1.00      0.96        11
           2       1.00      0.92      0.96        12

    accuracy                           0.97        30
   macro avg       0.97      0.97      0.97        30
weighted avg       0.97      0.97      0.97        30
```

In [15]:
```
y_test_array = np.ravel(y_test)
print(y_test_array)
```

```
[0 1 2 1 2 2 1 2 1 2 2 0 1 0 2 0 0 2 2 2 2 0 2 1 1 1 1 1 0 1]
```

In [16]:
```
print(y_pred)
```

```
[0 1 2 1 2 2 1 2 1 2 2 0 1 0 2 0 0 2 2 2 1 0 2 1 1 1 1 1 0 1]
```