In [1]:
```
import numpy as np
import pandas as pd
```

In [2]:
```
url = "https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data
names = ['sepal-length', 'sepal-width', 'petal-length', 'petal-width', 'Class'
df = pd.read_csv(url, names=names)
```

In [3]:
```
df.head()
```

Out[3]:

	sepal-length	sepal-width	petal-length	petal-width	Class
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

In [4]:
```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
df['Class'] = le.fit_transform(df['Class'])
```

In [5]:
```
x = df.drop("Class", axis=1)
```

In [6]:
```
y = df['Class']
```

In [7]:
```
from sklearn.model_selection import train_test_split
x_train,x_test, y_train,y_test = train_test_split(x,y, test_size = 0.3, random_
```

In [8]:
```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
x_train = sc.fit_transform(x_train)
x_test = sc.transform(x_test)
```

In [9]:
```
from sklearn.decomposition import PCA

pca = PCA(n_components=1)
x_train = pca.fit_transform(x_train)
x_test = pca.transform(x_test)
```

In [10]:
```
explained_variance = pca.explained_variance_ratio_
```

In [11]:
```
from sklearn.ensemble import RandomForestClassifier

classifier = RandomForestClassifier(max_depth=2, random_state=42)
classifier.fit(x_train, y_train)

# Predicting the Test set results
y_pred = classifier.predict(x_test)
```

In [12]:
```
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score

cm = confusion_matrix(y_test, y_pred)
print(cm)
print('Accuracy' , accuracy_score(y_test, y_pred))
```

```
[[19  0  0]
 [ 0 10  3]
 [ 0  0 13]]
Accuracy 0.9333333333333333
```

with 2 components

In [13]:
```
from sklearn.model_selection import train_test_split
x_train,x_test, y_train,y_test = train_test_split(x,y, test_size = 0.3, random
```

In [14]:
```
from sklearn.decomposition import PCA

pca = PCA(n_components=2)
x_train = pca.fit_transform(x_train)
x_test = pca.transform(x_test)
```

In [15]:
```
from sklearn.ensemble import RandomForestClassifier

classifier = RandomForestClassifier(max_depth=2, random_state=42)
classifier.fit(x_train, y_train)

# Predicting the Test set results
y_pred = classifier.predict(x_test)
```

In [16]:
```
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score

cm = confusion_matrix(y_test, y_pred)
print(cm)
print('Accuracy' , accuracy_score(y_test, y_pred))
```

```
[[19  0  0]
 [ 0 11  2]
 [ 0  0 13]]
Accuracy 0.9555555555555556
```

In [16]: