

In [17]: Aim: To Implement Random forest ensemble technique

```
In [ ]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import confusion_matrix, accuracy_score
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn import metrics
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

In [3]: dataset = pd.read\_csv('Logistic\_Iris.csv')

```
In [4]: x = dataset.iloc[:, [0,1,2,3]].values
y = dataset.iloc[:, 4].values
```

```
In [5]: xtrain, xtest, ytrain, ytest = train_test_split(x, y, test_size=0.25, random_state=0)
```

```
In [6]: sc = StandardScaler()
xtrain = sc.fit_transform(xtrain)
xtest = sc.transform(xtest)
```

```
In [8]: dtree= DecisionTreeClassifier()
dtree.fit(xtrain, ytrain)
```

```
Out[8]: ▼ DecisionTreeClassifier
DecisionTreeClassifier()
```

```
In [9]: y_pred1 = dtree.predict(xtest)
print("Predicted values:")
y_pred1
```

Predicted values:

```
Out[9]: array(['Iris-virginica', 'Iris-versicolor', 'Iris-setosa',
               'Iris-virginica', 'Iris-setosa', 'Iris-virginica', 'Iris-setosa',
               'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
               'Iris-virginica', 'Iris-versicolor', 'Iris-versicolor',
               'Iris-versicolor', 'Iris-versicolor', 'Iris-setosa',
               'Iris-versicolor', 'Iris-versicolor', 'Iris-setosa', 'Iris-setosa',
               'Iris-virginica', 'Iris-versicolor', 'Iris-setosa', 'Iris-setosa',
               'Iris-virginica', 'Iris-setosa', 'Iris-setosa', 'Iris-versicolor',
               'Iris-versicolor', 'Iris-setosa', 'Iris-virginica',
               'Iris-versicolor', 'Iris-setosa', 'Iris-virginica',
               'Iris-virginica', 'Iris-versicolor', 'Iris-setosa',
               'Iris-virginica'], dtype=object)
```

```
In [10]: acc_dtree= accuracy_score(ytest,y_pred1)*100
print ("\n\nAccuracy using Single Decision Tree: ", acc_dtree)
```

Accuracy using Single Decision Tree: 97.36842105263158

```
In [11]: cm = confusion_matrix(ytest, y_pred1)
print ("\n\n Confusion Matrix for Single Decision Tree: \n", cm)
```

Confusion Matrix for Single Decision Tree:

```
[[13  0  0]
 [ 0 15  1]
 [ 0  0  9]]
```

```
In [12]: RF = RandomForestClassifier(n_estimators = 100)
# Train the model using the training sets
RF.fit(xtrain, ytrain)
```

```
Out[12]: ▼ RandomForestClassifier
RandomForestClassifier()
```

```
In [13]: y_pred2 = RF.predict(xtest)
print("Predicted values:")
y_pred2
```

Predicted values:

```
Out[13]: array(['Iris-virginica', 'Iris-versicolor', 'Iris-setosa',
'Iris-virginica', 'Iris-setosa', 'Iris-virginica', 'Iris-setosa',
'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
'Iris-virginica', 'Iris-versicolor', 'Iris-versicolor',
'Iris-versicolor', 'Iris-versicolor', 'Iris-setosa',
'Iris-versicolor', 'Iris-versicolor', 'Iris-setosa', 'Iris-setosa',
'Iris-virginica', 'Iris-versicolor', 'Iris-setosa', 'Iris-setosa',
'Iris-virginica', 'Iris-setosa', 'Iris-setosa', 'Iris-versicolor',
'Iris-versicolor', 'Iris-setosa', 'Iris-virginica',
'Iris-versicolor', 'Iris-setosa', 'Iris-virginica',
'Iris-virginica', 'Iris-versicolor', 'Iris-setosa',
'Iris-virginica'], dtype=object)
```

```
In [15]: acc_rf= accuracy_score(ytest,y_pred2)*100
print ("\n\nAccuracy using Random Forest: ", acc_rf)
```

Accuracy using Random Forest: 97.36842105263158

```
In [16]: cm = confusion_matrix(ytest, y_pred2)
print ("\n\n Confusion Matrix for Random Forest Classifier: \n", cm)
```

```
Confusion Matrix for Random Forest Classifier:
[[13  0  0]
 [ 0 15  1]
 [ 0  0  9]]
```