

# Implement a Simple Decision Tree Classifier using Scikit Learn

## Importing required laibraries and datasets

```
In [16]: import pandas as pd
import numpy as np
from sklearn.datasets import load_iris
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
data=load_iris()
print('Classes to predict:',data.target_names)
```

Classes to predict: ['setosa' 'versicolor' 'virginica']

## Storing the dependant and independent attributes in seperate variables

```
In [31]: X=data.data
y=data.target
print("Number of records in dataset:",X.shape[0])
print(X[:4])
```

Number of records in dataset: 150  
[[5.1 3.5 1.4 0.2]  
 [4.9 3. 1.4 0.2]  
 [4.7 3.2 1.3 0.2]  
 [4.6 3.1 1.5 0.2]]

## Splitting training and test dataset seperately

```
In [32]: X_train,X_test,y_train, y_test=train_test_split
(X,y,random_state=22, test_size=0.45)
```

## Initializing the DecisionTreeClassifier with entropy as splitting metrics

```
In [33]: clf=DecisionTreeClassifier(criterion='entropy')
```

```
In [34]: clf.fit(X_train,y_train)
```

```
Out[34]: DecisionTreeClassifier(criterion='entropy')
```

```
In [35]: y_pred=clf.predict(X_test)
```

```
In [36]: from sklearn.metrics import accuracy_score
print('Accuracy score on train data', accuracy_score
      (y_true=y_train, y_pred=clf.predict(X_train)))
print('Accuracy score on test data', accuracy_score
      (y_true=y_test, y_pred=y_pred))
```

Accuracy score on train data 1.0  
Accuracy score on test data 0.9264705882352942

## Printing the Decision tree

```
In [39]: from sklearn.tree import DecisionTreeClassifier, plot_tree
plt.figure(figsize = (20,20))
plot_tree(clf, filled=True)
plt.show()
```

