

Module 4 : Implementing Decision Tree classifier

```
# Import the necessary packages.
```

```
import pandas as pd
```

```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

```
import seaborn as sns
```

In []:

```
import warnings
```

```
warnings.filterwarnings('ignore')
```

In []:

```
data = pd.read_csv('brain_stroke.csv')
```

In []:

```
data.head()
```

In []:

```
df = data.dropna()
```

In []:

```
from sklearn.preprocessing import LabelEncoder
```

```
le = LabelEncoder()
```

```
var = ['gender','smoking_status']
```

```
for i in var:
```

```
    df[i] = le.fit_transform(df[i]).astype(int)
```

In []:

```
df.columns
```

In []:

```
df.head()
```

In []:

```
df['stroke'].unique()
```

In []:

```
del df['ever_married']
```

```
del df['work_type']
```

```
del df['Residence_type']
```

In []:

```
df.head()
```

In []:

```
#preprocessing, split test and dataset, split response variable
```

```
X = df.drop(labels='stroke', axis=1)
```

```
#Response variable  
Y = df.loc[:, 'stroke']
```

In []:

```
import imblearn  
from imblearn.over_sampling import RandomOverSampler  
from collections import Counter  
  
ros = RandomOverSampler(random_state=1)  
x_ros, y_ros = ros.fit_resample(X, Y)  
print("OUR DATASET COUNT      : ", Counter(Y))  
print("OVER SAMPLING DATA COUNT : ", Counter(y_ros))
```

In []:

```
from sklearn.model_selection import train_test_split  
x_train, x_test, y_train, y_test = train_test_split(x_ros, y_ros, test_size=0.30,  
random_state=1, stratify=y_ros)  
print("Number of training dataset : ", len(x_train))  
print("Number of test dataset     : ", len(x_test))  
print("Total number of dataset    : ", len(x_train)+len(x_test))  
Implementing the DecisionTree classifier
```

In []:

```
from sklearn.metrics import confusion_matrix, classification_report,  
accuracy_score, plot_confusion_matrix  
from sklearn.tree import DecisionTreeClassifier
```

In []:

```
Dt = DecisionTreeClassifier()  
Dt.fit(x_train, y_train)  
predictDt = Dt.predict(x_test)  
Finding the accuracy of Decision Tree classifier.
```

In []:

```
accuracy = accuracy_score(y_test, predictDt)  
print("Accuracy of DecisionTree classifier:", accuracy * 100)  
Finding the classification Report Decision tree classifier.
```

In []:

```
cr = classification_report(y_test, predictDt)  
print("Classification report \n\n:", cr)  
Finding the Confusion matrix Decision tree classifier.
```

In []:

```
cm = confusion_matrix(y_test, predictDt)  
print("Confusion matrix:\n", cm)
```

In []:

```
import matplotlib.pyplot as plt
fig, ax = plt.subplots(figsize=(6,6))
plot_confusion_matrix(Dt, x_test, y_test, ax=ax)
plt.title('Confusion matrix of Decision Tree classifier')
plt.show()
```

In []:

```
import joblib
joblib.dump(Dt, 'Dt.pkl')
```

In []:

In []: