

```
In [3]: import numpy as np  
import pandas as pd
```

```
In [4]: data=pd.read_csv(r"D:\ML_Course\Works_on_python\Decision tree & Random Forest Calssification\diabetes.csv")
```

```
In [5]: data.head()
```

Out[5]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Diabetes	
0	6	148	72	35	0	33.6		0.627	50	pos
1	1	85	66	29	0	26.6		0.351	31	neg
2	8	183	64	0	0	23.3		0.672	32	pos
3	1	89	66	23	94	28.1		0.167	21	neg
4	0	137	40	35	168	43.1		2.288	33	pos

```
In [6]: data.isnull().any()
```

Out[6]:

```
Pregnancies      False  
Glucose          False  
BloodPressure    False  
SkinThickness    False  
Insulin          False  
BMI              False  
DiabetesPedigreeFunction  False  
Age              False  
Diabetes         False  
dtype: bool
```

```
In [8]: from sklearn.preprocessing import LabelEncoder  
le=LabelEncoder()  
data["Diabetes"]=le.fit_transform(data["Diabetes"])
```

```
In [9]: data.head()
```

Out[9]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Diabetes	
0	6	148	72	35	0	33.6		0.627	50	1
1	1	85	66	29	0	26.6		0.351	31	0
2	8	183	64	0	0	23.3		0.672	32	1
3	1	89	66	23	94	28.1		0.167	21	0
4	0	137	40	35	168	43.1		2.288	33	1

```
In [10]: x=data.iloc[:,0:8].values  
y=data.iloc[:,8:9].values
```

```
In [11]: x.shape
```

Out[11]: (768, 8)

```
In [17]: from sklearn.model_selection import train_test_split  
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=0)
```

```
In [18]: from sklearn.preprocessing import StandardScaler  
sc=StandardScaler()  
x_train=sc.fit_transform(x_train)  
x_test=sc.fit_transform(x_test)
```

## DecisionTreeClassifier

```
In [20]: from sklearn.tree import DecisionTreeClassifier  
dtc=DecisionTreeClassifier(criterion="entropy",random_state=0)  
dtc.fit(x_train,y_train)
```

```
Out[20]: DecisionTreeClassifier(class_weight=None, criterion='entropy', max_depth=None,  
max_features=None, max_leaf_nodes=None,  
min_impurity_decrease=0.0, min_impurity_split=None,  
min_samples_leaf=1, min_samples_split=2,  
min_weight_fraction_leaf=0.0, presort=False, random_state=0,  
splitter='best')
```

```
In [21]: y_predict=dtc.predict(x_test)
```

```
In [22]: y_predict
```

```
Out[22]: array([1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0,  
0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1,  
1, 0, 0, 0, 1, 0, 1, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 1, 1,  
1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0,  
1, 0, 1, 1, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 0, 1, 0, 1,  
0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0,  
0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0])
```

```
In [23]: from sklearn.metrics import accuracy_score  
accuracy_score(y_test,y_predict)
```

```
Out[23]: 0.7012987012987013
```

```
In [31]: from sklearn.metrics import confusion_matrix  
cm=confusion_matrix(y_test,y_predict)
```

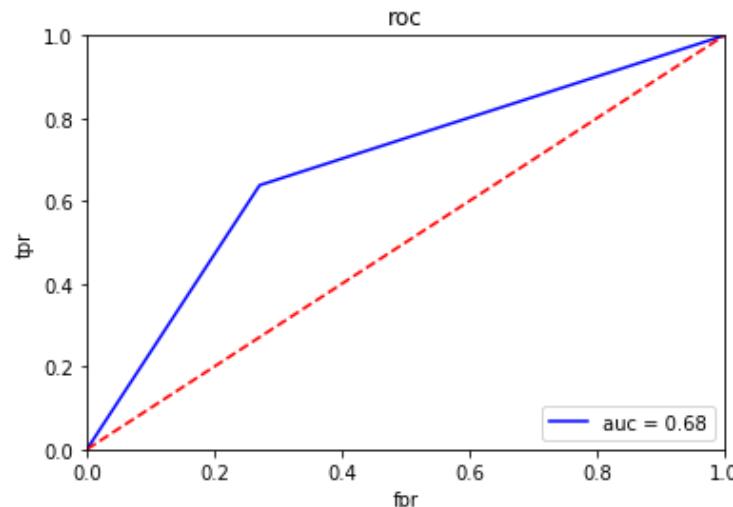
```
In [32]: cm
```

```
Out[32]: array([[78, 29],  
[17, 30]], dtype=int64)
```

```
In [33]: import sklearn.metrics as metrics  
fpr,tpr,threshold=metrics.roc_curve(y_test,y_predict)  
roc_auc=metrics.auc(fpr,tpr)
```

```
In [34]: import matplotlib.pyplot as plt
plt.title("roc")
plt.plot(fpr,tpr,'b',label = 'auc = %0.2f'%roc_auc)
plt.legend(loc = 'lower right')
plt.plot([0,1],[0,1],'r--')
plt.xlim([0,1])
plt.ylim([0,1])
plt.ylabel('tpr')
plt.xlabel('fpr')
```

```
Out[34]: Text(0.5, 0, 'fpr')
```



```
In [35]: dtc.predict(sc.transform([[1,150,80,24,33,46,4,66]]))
```

```
Out[35]: array([1])
```

```
In [37]: data.head(1)
```

```
Out[37]:
```

Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Diabetes
0	6	148	72	35	0	33.6	0.627	50

# RandomForestClassifier

```
In [258]: from sklearn.ensemble import RandomForestClassifier  
rfc = RandomForestClassifier(n_estimators = 1000,criterion = 'entropy',random_state = 0)
```

```
In [259]: rfc.fit(x_train,y_train)
```

C:\Users\anikp\Anaconda3\lib\site-packages\ipykernel\_launcher.py:1: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n\_samples,), for example using ravel().

"""Entry point for launching an IPython kernel.

```
Out[259]: RandomForestClassifier(bootstrap=True, class_weight=None, criterion='entropy',  
                                 max_depth=None, max_features='auto', max_leaf_nodes=None,  
                                 min_impurity_decrease=0.0, min_impurity_split=None,  
                                 min_samples_leaf=1, min_samples_split=2,  
                                 min_weight_fraction_leaf=0.0, n_estimators=1000, n_jobs=None,  
                                 oob_score=False, random_state=0, verbose=0, warm_start=False)
```

```
In [260]: y_pred1 = rfc.predict(x_test)
```

```
In [261]: y_pred1
```

```
Out[261]: array([1, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0,  
0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1,  
1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1,  
1, 0, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,  
1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 1,  
0, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0,  
0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0])
```

In [262]: y\_test

```
In [263]: from sklearn.metrics import accuracy_score  
accuracy_score(y_pred1,y_test)
```

**Out[263]:** 0.8181818181818182

```
In [264]: from sklearn.metrics import confusion_matrix  
cm1 = confusion_matrix(y_test,y_pred1)
```

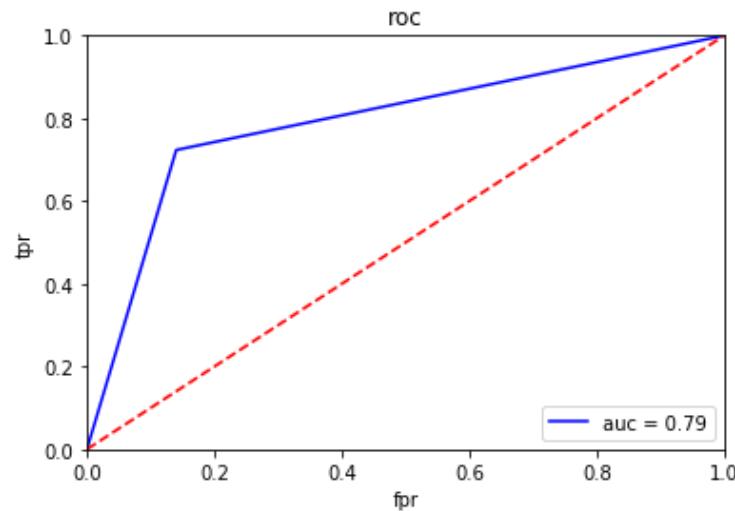
In [265]: cm1

```
Out[265]: array([[92, 15],  
                  [13, 34]], dtype=int64)
```

```
In [266]: import sklearn.metrics as metrics  
fpr1,tpr1 , threshold = metrics.roc_curve(y_test,y_pred1)  
roc_auc1 = metrics.auc(fpr1,tpr1)
```

```
In [267]: plt.title("roc")
plt.plot(fpr1,tpr1,'b',label = 'auc = %0.2f'%roc_auc1)
plt.legend(loc = 'lower right')
plt.plot([0,1],[0,1],'r--')
plt.xlim([0,1])
plt.ylim([0,1])
plt.ylabel('tpr')
plt.xlabel('fpr')
```

Out[267]: Text(0.5, 0, 'fpr')



In [ ]: