M.S.SANJAY

15BCE0517

L7+L8

DECISION TREE

DATASET:

|  |  |
| --- | --- |
| |  | | --- | | **Blood Transfusion Service Center Data Set** | |

Given is the variable name, variable type, the measurement unit and a brief description. The "Blood Transfusion Service Center" is a classification problem. The order of this listing corresponds to the order of numerals along the rows of the database.   
  
R (Recency - months since last donation),   
F (Frequency - total number of donation),   
M (Monetary - total blood donated in c.c.),   
T (Time - months since first donation), and   
a binary variable representing whether he/she donated blood in March 2007 (1 stand for donating blood; 0 stands for not donating blood).

CODE:

import numpy as np

import pandas as pd

from sklearn.metrics import confusion\_matrix

from sklearn.cross\_validation import train\_test\_split

from sklearn.tree import DecisionTreeClassifier

from sklearn.metrics import accuracy\_score

from sklearn.metrics import classification\_report

def importdata():

balance\_data = pd.read\_csv(

'https://archive.ics.uci.edu/ml/machine-learning-databases/blood-transfusion/transfusion.data',

sep= ',', header = 0 )

print ("Dataset Lenght: ", len(balance\_data))

print ("Dataset Shape: ", balance\_data.shape)

print ("Dataset: ",balance\_data.head())

return balance\_data

def splitdataset(balance\_data):

X = balance\_data.values[:, 0:4]

Y = balance\_data.values[:, 4]

X\_train, X\_test, y\_train, y\_test = train\_test\_split(

X, Y, test\_size = 0.3, random\_state = 100)

return X, Y, X\_train, X\_test, y\_train, y\_test

def train\_using\_gini(X\_train, X\_test, y\_train):

clf\_gini = DecisionTreeClassifier(criterion= "gini",

random\_state = 100,max\_depth=3, min\_samples\_leaf=5)

clf\_gini.fit(X\_train, y\_train)

return clf\_gini

def tarin\_using\_entropy(X\_train, X\_test, y\_train):

clf\_entropy = DecisionTreeClassifier(

criterion = "entropy", random\_state = 100,

max\_depth = 3, min\_samples\_leaf = 5)

clf\_entropy.fit(X\_train, y\_train)

return clf\_entropy

def prediction(X\_test, clf\_object):

y\_pred = clf\_object.predict(X\_test)

print("Predicted values:")

print(y\_pred)

return y\_pred

def cal\_accuracy(y\_test, y\_pred):

print("Confusion Matrix: ",

confusion\_matrix(y\_test, y\_pred))

print ("Accuracy : ",

accuracy\_score(y\_test,y\_pred)\*100)

print("Report : ",

classification\_report(y\_test, y\_pred))

def main():

data = importdata()

X, Y, X\_train, X\_test, y\_train, y\_test = splitdataset(data)

clf\_gini = train\_using\_gini(X\_train, X\_test, y\_train)

clf\_entropy = tarin\_using\_entropy(X\_train, X\_test, y\_train)

print("Results Using Gini Index:")

y\_pred\_gini = prediction(X\_test, clf\_gini)

cal\_accuracy(y\_test, y\_pred\_gini)

print("Results Using Entropy:")

y\_pred\_entropy = prediction(X\_test, clf\_entropy)

cal\_accuracy(y\_test, y\_pred\_entropy)

if \_\_name\_\_=="\_\_main\_\_":

main()

***OUTPUT:***

Dataset Lenght: 748

Dataset Shape: (748, 5)

Dataset: Recency (months) Frequency (times) Monetary (c.c. blood) Time (months) \

0 2 50 12500 98

1 0 13 3250 28

2 1 16 4000 35

3 2 20 5000 45

4 1 24 6000 77

whether he/she donated blood in March 2007

0 1

1 1

2 1

3 1

4 0

Results Using Gini Index:

Predicted values:

[1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0

0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 1 0 1 0 0 1

0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0

0 1 0 1 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0

0 1 0]

Confusion Matrix: [[159 10]

[ 37 19]]

Accuracy : 79.1111111111

Report : precision recall f1-score support

0 0.81 0.94 0.87 169

1 0.66 0.34 0.45 56

avg / total 0.77 0.79 0.77 225

Results Using Entropy:

Predicted values:

[1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0

0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 1 0 1 0 0 1

0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0

0 1 0 1 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0

0 1 0]

Confusion Matrix: [[159 10]

[ 37 19]]

Accuracy : 79.1111111111

Report : precision recall f1-score support

0 0.81 0.94 0.87 169

1 0.66 0.34 0.45 56

avg / total 0.77 0.79 0.77 225

>>>

**SCREENSHOT:**



