1		EXP NO : 13 - DECIDION TREE CLASSIFICATION
1		AIM! vote instruction of trapert total most tale move
1	(0= 3)	to classify the social Network dataset using recision tree analysis.
1		COUNTY - COND - A DECEMBER OF THE CONTRACT OF
		ALCHORATION: (12012) Cillary robbinols a long-
	1)	Ampoint Librarias and Local the alotocet
		simon mathem anger contentence of a party
1	2)	Define x (footwee) and y (lobele).
1		(ma) Arting
1	3)	split data into training and test sets (75% train, 25% test).
1		mond - p. ming x = 10 - y, 100 - x
T	4)	Scale teatures with standard scales.
T		1 - Connection of the x = thought appearance on birefluor and = 100 pt
Ť	5)	Anihialize and trouin pecision Tracclassificar using entropy
1		(((C=0 = K04) ,) + () x0m - () (140) - x = q=0 ()
1	12	product on test date and generate a compaison mothix.
t	67	PLE-LEMBERGH CX1, X2, classifier, parchet capearay CIX, as ford (),
r		
t	01.00000000	propore x-set and y-set for visuelization.
t		
H		CODE: (C) KEDNIGK , (Snice alk Smilk allq
H		import pordos as pd (Oxom-sx Oxom-sx) an
H		import numpy (as inpusion go) standard of its
		simport motpholib pyphot as put
L		datoset = pol. 1000 - csv ("I content / galinia / my Drive / social - Network - Ada cov
L		
		x = dotorat.iloc [:, [2/3]]. value
		y = dotoset.iloc (:,-1). volue
		trom skloann model-soloction import train-text-split.
		trom sklourn model-soloction import train-text-split.
		trom skloann andal-soloction import train-text-split. x-train x-test, y-train, y-test = train-test-split (x,y, test-size = 0.25,
		troin skloom model-soloction import train-text-split. x-train, x-test, y-train, y-test= train-test-split (x,y, test-size = 0.25, random-state = 0)
		trom skloom model-soloction import train-text-split. x-train, x-test, y-train, y-test= train-test-split (x,y, test-size=0.25, random-state=0) trum skloom proprocessing import standardscolor
		troin skloom model-soloction import train-text-split. x-train, x-test, y-train, y-test= train-test-split (x,y, test-size = 0.25, random-state = 0)

x-test = sc-trongprom (x-tost).

from alcloom trae import pacision Trae classifier
elassifier = poession Tracclassifier (criterion = 'entropy', rondom-state :0)
classifier. fit (x-train, y-train)
y-prod = classifier. prodict (x-text)

from elelearn motorice import confusion-motorix

cm = confusion-motorix (y-test, y-pred)

print (cm)
from Cinatplotlib. to colons impost listedColormap

X-set, y-set = x-train, y-train

x1 / x2 = np. meshgrid (np. armonge (stant = x-99t [:107.min()-1,

stop = x-00t [:10]. max()+1, stop= 0.01), np. armonge (stort = x-00t [:1]=

1, stop = x-00t [:1]. max()+1, stop= 0.01).

plt-contourt (x1, x2, clossifier. prodict (np. array ([x1, ravel(), x2, radio))
roshaps (x1. shape), alpha = 0.75, cmop = listed colormae (c'radigna)

plt. ylim(x2·min(), x1·max())

plt. ylim(x2·min(), x2·max())

for i, j in gnumerate (np. unique cy-set)):

put scotter (x-set (y-set == j,0], x-set (y-set == j,1] = c= listed colormop (cared , green) (i), lobel = j)

plt-title ('Docision Times classification (Training ser+1)

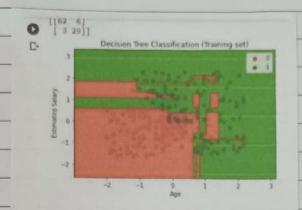
pt. ylabet ('purchase')

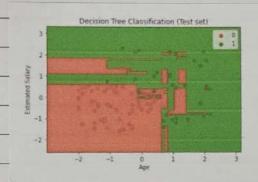
plt. Logard ()

makered of 1



OUTPUT :





PSOUT:

REQUUT:

Thus pacinion tree analysis is successfully executed and