**Codes in Python**

#importing libraries

import os

os.chdir("C:/Python27/Lib")

os.getwd()

impot pandas as pd

import numpy as np

import matplotlib as mtl

from Sklear.ensemble import RandomForestClassifier

from Sklearn import tree

from Sklearn.metrices import accuracy\_Score

import Sklear.cross\_validation import train\_test\_split

from Sklearn.tree import DecissiontreeRegressor

#loading data

customer = pd.read\_csv(“customer.csv” , sep = “,”)

#removing the 1 variable: “ID\_Code”

del ID\_Code

#creating dataframe only for missing values

Missing – pd.Date.frame(customer.insull().sum())

#Feature selection

#feature scaling

%matplotlib.inline

Plt.hist(cab [“target”] , bins = ‘auto’)

cnames = [ "var\_99", "var\_170"]

for i in cnames

customer [,i] = (customer[,i] - min(customer[,i]))/ (max(customer[,i] - min(customer[,i])))

#sample

x = customer.value[:,1:2]

y = customer.value[,0]

x\_train, x-test, y\_train, y\_test = train\_test

split(x,y, test\_size = 0.2)

#decision tree

Clf = tree.DecisiontreeClassifier (criterion = ‘entropy’). Fit(x\_train, y\_train)

Y\_pred = clf.predict(x\_test)

Dotfile = open(“pt.dot”.”w”)

Df = tree.export\_graphing(clf,outfile, feature\_names = customer.columns, feature\_names = customer.columns)

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

#89.3%

#Rondom Forest

From sklearn.ensemble import RondomForestClassifier

RF = RondomForestClassifier(n\_estimators = 100)

Fit(x\_train, y\_train)

RF = RF.modelPredict(x\_test)

#building confusion matrix

From sklearn.metrics import confusion\_matrix

Cm = confusion\_matrix(y\_test, y\_ped)

Cm = pd.crosstable(y\_test, RF\_ped)

Accuracy\_score = (y\_test, y\_ped)\*100

88.4%