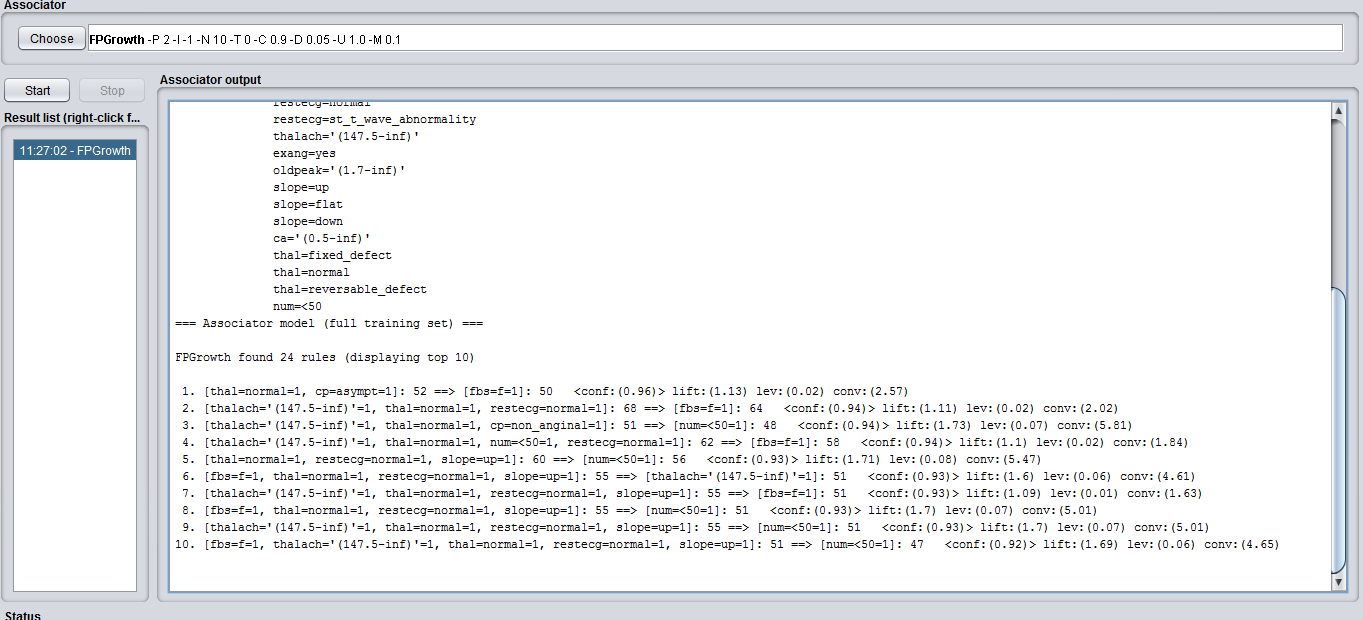
DMWA Lab – 5

Q1

FP\_growth Rule in weka



Q2

import pandas as pd  
import numpy as np

import pyfpgrowth

df= pd.read\_csv(“ transaction\_data.csv”)

patterns = pyfpgrowth. find\_frequent\_patterns(transactions, 10)

rules = pyfpgrowth. generate\_association\_rules(patterns,0.8)

def support\_count(rhs):

count=0

rhs= set(rhs)

for j in df[‘items’]:

j=set(j)

if(rhs.issubset(j)):

count=count+1

return count

rhs\_support = []

for I in rules\_df[‘Consequent’]:

a=support\_count(i)

rhs\_support.append(a/len(df))

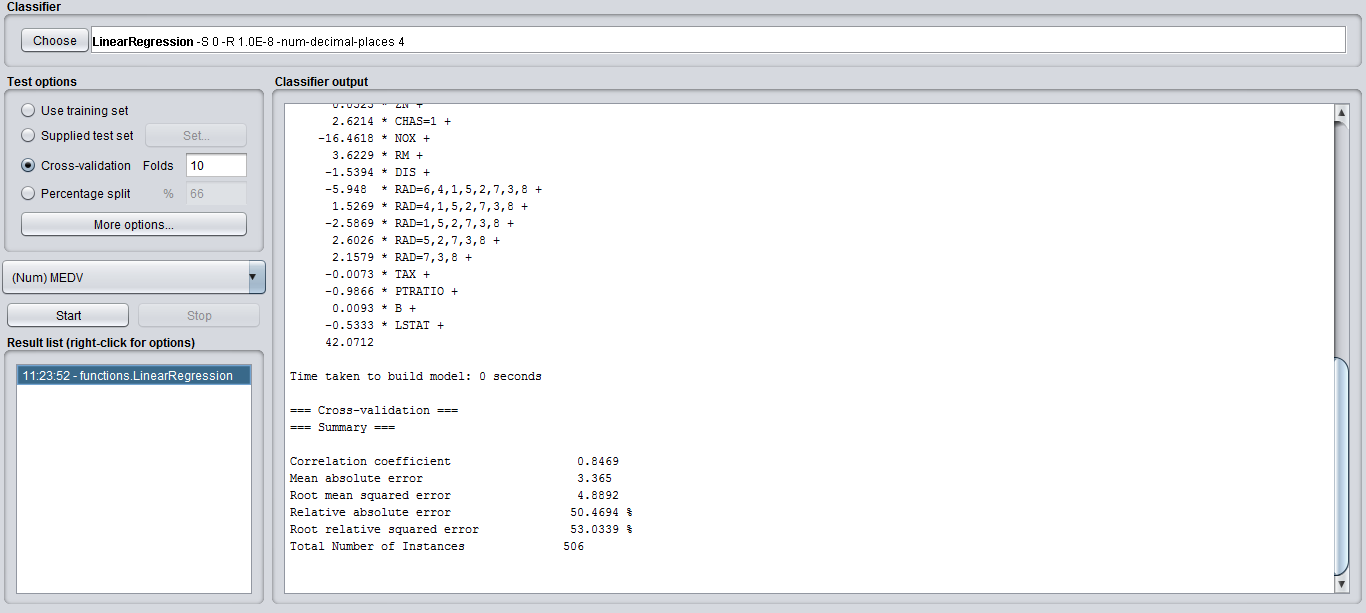
rules\_df[‘RHS\_support’] = rhs\_support

rules\_df[‘lift’] = rules\_df[‘Confidence’]/rules\_df[‘RHS\_support’]

rules\_df[‘Conviction’] = (1-rules\_df[‘RHS\_support’])/(1-rules\_df[‘Confidence’])

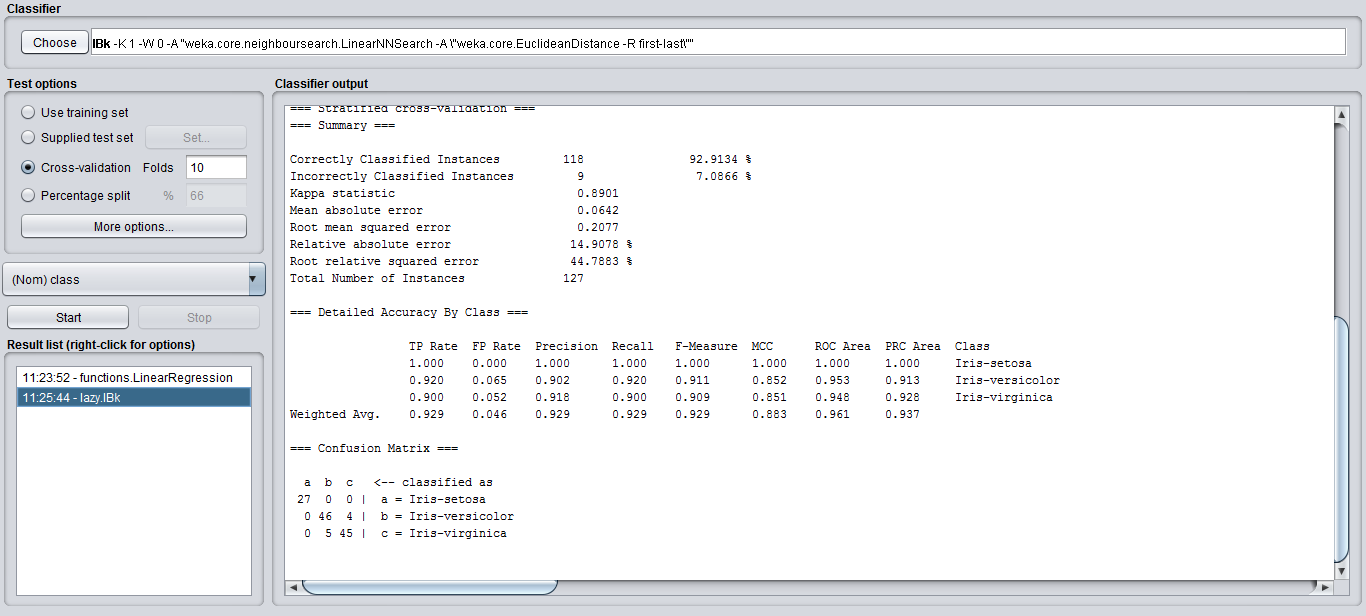
Q3

Linear Regression using Weka



Q4

KNN using Weka on IRIS dataset



Code-:

from math import sqrt  
  
def euclidean\_distance(row1, row2):  
distance = 0.0  
for i in range(len(row1)-1):  
distance += (row1[i] - row2[i])\*\*2  
return sqrt(distance)  
  
def get\_neighbors(train, test\_row, num\_neighbors):  
distances = list()  
for train\_row in train:  
dist = euclidean\_distance(test\_row, train\_row)  
distances.append((train\_row, dist))  
distances.sort(key=lambda tup: tup[1])  
neighbors = list()  
for i in range(num\_neighbors):  
neighbors.append(distances[i][0])  
return neighbors

def predict\_classification(train, test\_row, num\_neighbors):  
neighbors = get\_neighbors(train, test\_row, num\_neighbors)  
output\_values = [row[-1] for row in neighbors]  
prediction = max(set(output\_values), key=output\_values.count)  
return prediction