**Name: Adithya M Section: K SRN: PES1UG20CS621**

"""

Assume df is a pandas dataframe object of the dataset given

"""

import numpy as np

import pandas as pd

import random

"""Calculate the entropy of the enitre dataset"""

# input:pandas\_dataframe

# output:int/float

def get\_entropy\_of\_dataset(df):

entropy = 0

column\_values = df[[df.columns[-1]]].values

a, unique\_count = np.unique(column\_values, return\_counts=True)

no\_of\_instances = len(column\_values)

if no\_of\_instances <= 1:

return 0

probs\_array = []

for i in range(0, len(unique\_count)):

probs = unique\_count[i] / no\_of\_instances

probs\_array.append(probs)

for probabilities in probs\_array:

if probabilities != 0:

entropy = entropy - (probabilities \* np.log2(probabilities))

return entropy

"""Return avg\_info of the attribute provided as parameter"""

# input:pandas\_dataframe,str {i.e the column name ,ex: Temperature in the Play tennis dataset}

# output:int/float

def get\_avg\_info\_of\_attribute(df, attribute):

avg\_info\_of\_attribute = 0

attribute\_values = df[attribute].values

unique\_attribute\_values, unique\_attribute\_array = np.unique(

attribute\_values, return\_counts=True

)

no\_of\_instances = len(attribute\_values)

for attribute\_value in unique\_attribute\_values:

sliced\_dataframe = df[df[attribute] == attribute\_value]

instances = sliced\_dataframe[[sliced\_dataframe.columns[-1]]].values

instances\_unique\_values, instances\_unique\_counts = np.unique(

instances, return\_counts=True

)

total\_count\_in\_an\_instance = len(instances)

entropy\_of\_attribute\_value = 0

for i in instances\_unique\_counts:

j = i / total\_count\_in\_an\_instance

if j != 0:

entropy\_of\_attribute\_value = entropy\_of\_attribute\_value - (

j \* np.log2(j)

)

avg\_info\_of\_attribute = avg\_info\_of\_attribute + entropy\_of\_attribute\_value \* (

total\_count\_in\_an\_instance / no\_of\_instances

)

return abs(avg\_info\_of\_attribute)

"""Return Information Gain of the attribute provided as parameter"""

# input:pandas\_dataframe,str

# output:int/float

def get\_information\_gain(df, attribute):

information\_gain = 0

entropy\_of\_dataset = get\_entropy\_of\_dataset(df)

entropy\_of\_attribute = get\_avg\_info\_of\_attribute(df, attribute)

information\_gain = entropy\_of\_dataset - entropy\_of\_attribute

return information\_gain

# input: pandas\_dataframe

# output: ({dict},'str')

def get\_selected\_attribute(df):

"""

Return a tuple with the first element as a dictionary which has IG of all columns

and the second element as a string with the name of the column selected

example : ({'A':0.123,'B':0.768,'C':1.23} , 'C')

"""

max\_information\_gain = 0

information\_gain\_of\_all\_attributes = {}

selected\_attribute = ""

for attribute in df.columns[:-1]:

information\_gain\_of\_an\_attribute = get\_information\_gain(df, attribute)

if information\_gain\_of\_an\_attribute > max\_information\_gain:

max\_information\_gain = information\_gain\_of\_an\_attribute

selected\_attribute = attribute

information\_gain\_of\_all\_attributes[attribute] = information\_gain\_of\_an\_attribute

return (information\_gain\_of\_all\_attributes, selected\_attribute)

Output

