Universidad Politécnica Salesiana

jarevalop1@est.ups.edu.ec

Jorge Arévalo

Las tareas para realizar son las siguientes:

- Reprocesar los datos del corpus de acuerdo con las sugerencias desarrolladas por wguillen [github].
- Aplicar la técnica de los vecinos más cercanos indicada en clase y empleando la fórmula propuesta por wguillen.
- Desarrollar una pequeña interfaz en Python u otro lenguaje donde se coloquen los atributos y el sistema indique la calidad del vino.
- Realizar un pequeño informe del trabajo desarrollado, considerando los aspectos principales y qué tan preciso es el sistema.

Reprocesar los datos del corpus de acuerdo con las sugerencias desarrolladas por wguillen [github].

Código en Python

```
from tkinter import *
from tkinter import ttk
from tkinter import messagebox
import pandas as pd
import operator
import csv
import os
rt = Tk()
def analizar():
    newWindows=Tk()
    newWindows.title("REPORTE")
    df = pd.read_csv('winequality-red.csv')
    wine = [list(row) for row in df.values]
    similares={}

cn=[float(tfa.get()),float(tva.get()),float(tca.get()),float(trs.get()),float(tc.get()),
float(tfsd.get()),

float(ttsd.get()),float(td.get()),float(tph.get()),float(ts.get()),float(ta.get())]
    minimo=[4.6,0.12,0,0.9,0.012,1,6,0.99,2.74,0.33,8.4]
    maximo=[15.9,1.58,1.0,13.9,0.611,72.0,289.0,1.0,4.01,2.0,14.9]

weight=[float(ctfa.get()),float(ctva.get()),float(ctca.get()),float(ctrs.get()),float(ct
c.get()),float(ctfsd.get()),

float(cttsd.get()),float(ctd.get()),float(ctph.get()),float(cts.get()),float(cta.get())]
    def similarity(ce):
        valor=0
        for i in range(len(minimo)):
            valor+= weight[i] * (1-((abs(float(ce[i])-cn[i]))/(maximo[i]-minimo[i])))
        return valor/sum(weight)

    for i in range(len(wine)):
        fila=[]
        fila=wine[i]
        x = similarity(fila)
        similares.update({str(i):round(x,3)})

    ordenados = dict(sorted(similares.items(), key=operator.itemgetter(1)))
    cols=("#Wine", "Fixed Acidity", "Volatile Acidity", "Citric Acid","Residual
Sugar","Chlorides","Free Sulfure Dioxide","Total Sulfure
Dioxide","Density","pH","Sulphates","Alcohol","Quality","Similarity")
    tree = ttk.Treeview(newWindows,columns=cols,show='headings')
    vsb = ttk.Scrollbar(newWindows, orient="vertical", command=tree.yview)
    vsb.pack(side=RIGHT, fill=BOTH)

    tree.configure(yscrollcommand=vsb.set)
    for i in range(len(cols)):
        tree.heading(cols[i],text=cols[i])
```

```
        tree.column(cols[i], minwidth=0, width=50)
    tree.pack(expand=YES, fill=BOTH)
    tam=len(ordenados)
    for i in range(tam):
        pos=int(list(ordenados.items())[i][0])
        c1=wine[int(pos)][0]
        c2=wine[int(pos)][1]
        c3=wine[int(pos)][2]
        c4=wine[int(pos)][3]
        c5=wine[int(pos)][4]
        c6=wine[int(pos)][5]
        c7=wine[int(pos)][6]
        c8=wine[int(pos)][7]
        c9=wine[int(pos)][8]
        c10=wine[int(pos)][9]
        c11=wine[int(pos)][10]
        c12=wine[int(pos)][11]
        sim=str(list(ordenados.items())[i][1])
        tree.insert("",0,i,values=(str(pos),c1,c2,c3,c4,c5,c6,c7,c8,c9,c10,c11,c12,sim))

    fpos=list(ordenados.items())[tam-1][0]
    fval=list(ordenados.items())[tam-1][1]
    res=wine[int(fpos)][11]

    li=[cn[0],cn[1],cn[2],cn[3],cn[4],cn[5],cn[6],cn[7],cn[8],cn[9],cn[10],res]
    if li in wine:
        messagebox.showinfo(message="Calificacion: " + res + "/n"+"Similitud:
"+str(fval))
    else:
        with open('winequality-red.csv','a') as f:
            writer = csv.writer(f)
            writer.writerow((li))
        messagebox.showinfo(message="Calificacion: " + res + "/n"+"Similitud:
"+str(fval))

array =[0,1,2,3,4,5,6,7,8,9,10]
rt.geometry('800x225')
rt.config(bg="green")
rt.title('winequality-red')
Label(rt,text="WINE-QUALITY",fg="red",font=("Comic Sans MC",15,'bold')).place(x=300,y=0)
Label(rt,text="Fixed Acidy",font=('bold')).place(x=0,y=25)
tfa = Spinbox(rt, from_=4.6, to=15.9 , width=5,increment=0.1,font='Helvetica 12')
tfa.place(x=150,y=25)
ctfa = ttk.Combobox(rt,values=array,width=5,font='Helvetica 12')
ctfa.place(x=225, y=25)
ctfa.current(3)

Label(rt,text="Volatily Acidy",font=('bold')).place(x= 308,y=25)
tva = Spinbox(rt, from_=0.12, to=1.58 , width=5,increment=0.01,font='Helvetica 12')
tva.place(x=450 ,y=25)
ctva = ttk.Combobox(rt,values=array,width=5,font='Helvetica 12')

ctva.place(x=525, y=25)

ctva.current(3)

Label(rt,text="Citric Acid",font=('bold')).place(x=0,y=50)

tca = Spinbox(rt, from_=0.0, to=1.0 , width=5,increment=0.1,font='Helvetica 12')

tca.place(x=150 ,y=50)

ctca = ttk.Combobox(rt,values=array,width=5,font='Helvetica 12')

ctca.place(x=225, y=50)

ctca.current(3)

Label(rt,text="Residual Sugar",font=('bold')).place(x= 308,y=50)

trs = Spinbox(rt, from_=0.9, to=13.9 , width=5,increment=0.1,font='Helvetica 12')

trs.place(x=450 ,y=50)

ctrs = ttk.Combobox(rt,values=array,width=5,font='Helvetica 12')

ctrs.place(x=525, y=50)
```

```
ctrs.current(5)
Label(rt,text="Chlorides",font=('bold')).place(x=0,y=75)
tc = Spinbox(rt, from_=0.012, to=0.611 , width=5,increment=0.001,font='Helvetica 12')
tc.place(x=150 ,y=75)
ctc = ttk.Combobox(rt,values=array,width=5,font='Helvetica 12')
ctc.place(x=225, y=75)
ctc.current(1)
Label(rt,text="Free Sulfur Dioxide",font=('bold')).place(x= 308,y=75)
tfsd = Spinbox(rt, from_=1.0, to=72.0 , width=5,increment=1.0,font='Helvetica 12')
tfsd.place(x=450 ,y=75)
ctfsd = ttk.Combobox(rt,values=array,width=5,font='Helvetica 12')
ctfsd.place(x=525, y=75)
ctfsd.current(1)
Label(rt,text="Total Sulfure Dioxide",font=('bold')).place(x=0,y=100)
ttsd = Spinbox(rt, from_=6.0, to=289.0 , width=5,increment=1,font='Helvetica 12')
ttsd.place(x=150 ,y=100)
cttsd = ttk.Combobox(rt,values=array,width=5,font='Helvetica 12')
cttsd.place(x=225, y=100)
cttsd.current(1)
Label(rt,text="Density",font=('bold')).place(x= 308,y=100)
td= Spinbox(rt, from_=0.9900, to=1.0000 , width=6,increment=0.0001,font='Helvetica 12')
td.place(x=450 ,y=100)
ctd = ttk.Combobox(rt,values=array,width=5,font='Helvetica 12')
ctd.place(x=525, y=100)
ctd.current(1)
Label(rt,text="pH",font=('bold')).place(x=0,y=125)
tph = Spinbox(rt, from_=2.74, to=4.01 , width=5,increment=0.01,font='Helvetica 12')
tph.place(x=150 ,y=125)
ctph = ttk.Combobox(rt,values=array,width=5,font='Helvetica 12')
ctph.place(x=225, y=125)
ctph.current(6)
Label(rt,text="Sulphates",font=('bold')).place(x= 308,y=125)
ts= Spinbox(rt, from_=0.33, to=2.0 , width=5,increment=0.01,font='Helvetica 12')
ts.place(x=450 ,y=125)
cts = ttk.Combobox(rt,values=array,width=5,font='Helvetica 12')
cts.place(x=525, y=125)
cts.current(1)
Label(rt,text="Alcohol",font=('bold')).place(x=0,y=150)
ta = Spinbox(rt, from_=8.4, to=14.9 , width=5,increment=0.1,font='Helvetica 12')
ta.place(x=150 ,y=150)
cta = ttk.Combobox(rt,values=array,width=5,font='Helvetica 12')
```

```
cta.place(x=225, y=150)

cta.current(5)

ttk.Button(rt, text='Generar Reporte', command=analizar).place(x=500,y=200)

rt.mainloop()
```

El único resultado proporcionado por RBC será la calidad del vino, dependiendo de los parámetros de entrada.

El cálculo de similitud entre el caso buscado y los casos en la base de conocimiento se realiza utilizando la fórmula:

La fórmula propuesta por wguillen.

$$Similaridade(A1C1, A1C2) = 1 - \frac{|A1C2 - A1C1|}{(val\ max - val\ min)}$$

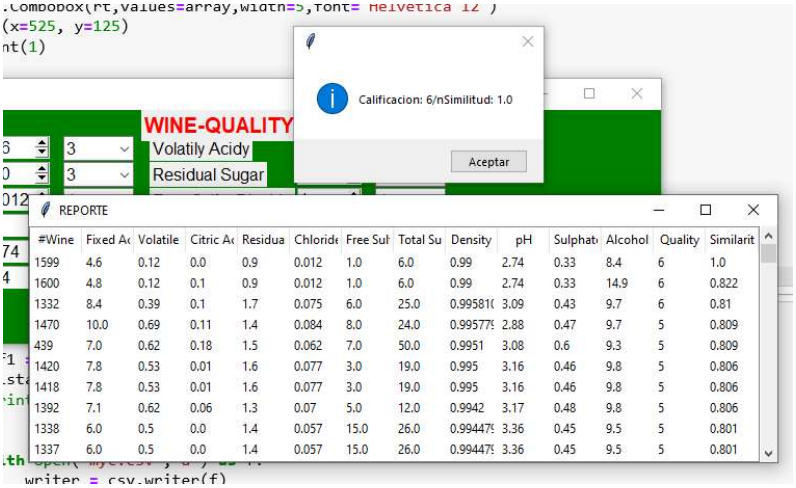
Intervalo de variação!

Formula en Python

```
  def similarity(ce):

        valor=0

        for i in range(len(minimo)):

            valor+= weight[i] * (1-((abs(float(ce[i])-cn[i]))/(maximo[i]-
minimo[i])))

        return valor/sum(weight)
```

Interfaz Python

Generar reporte



Nos muestra un reporte de la calidad del vino