

## INFORME DE SISTEMA DE VINOS

Nombre: Jordan Murillo

Fecha: Viernes, 22 de mayo de 2020

Como caso práctico se implementará un sistema CBR básico para determinar la calidad del vino rojo. Para ello, se trabajará con el corpus **Wine Quality Data Set**. El corpus se compone de un total de 1599 muestras de vino rojo que contienen información de pruebas fisicoquímicas realizadas en vinos rojos.

Para ello, se deberá considerar lo siguiente:

Se tienen los siguientes atributos del vino en archivo “winequality-red.csv” delimitado por comas. Tareas para realizar:

I. Preprocesar los datos del corpus de acuerdo con las sugerencias desarrolladas por wguillen [github]. Esto se llevará a cabo en el diseño del programa en Python.

1 - fixed acidity
2 - volatile acidity
3 - citric acid
4 - residual sugar
5 - chlorides
6 - free sulfur dioxide
7 - total sulfur dioxide
8 - density
9 - pH
10 - sulphates
11 - alcohol
Variable de salida:
12 - quality (puntaje entre 0 y 10)

II. Aplicar las técnicas de los vecinos más cercanos indicada en clase y empleando la fórmula propuesta por wguillen.

Esta información está el documento de Power Point:

Fórmula para calcular la similitud:

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

Máximos y Mínimos de los atributos:

fixed acidity

min → 4.6

max → 15.9

volatile acidity

min → 0.12

max → 1.58

citric acid

min → 0

max → 1

residual sugar

min → 0.9

max → 13.9

chlorides  
min→ 0.012  
max→ 0.611  
free sulfur dioxide  
min→ 1  
max→ 72  
total sulfur dioxide  
min→ 6  
max→ 289  
density  
min→ 0.99  
max→ 1  
pH  
min→ 2.74  
max→ 4.01  
sulphates  
min→ 0.33  
max→ 2  
alcohol  
min→ 8.4  
max→ 14.9

quality

Este atributo sera dado con el análisis de las características de cada vino en el archivo .csv.

III. Desarrollar una pequeña interfaz en Python u otro lenguaje donde se coloquen los atributos y el sistema indique la calidad del vino.

Código de Python

```
#Importar librerías necesarias
from tkinter import *      # Carga módulo tk (widgets estándar)
from tkinter import ttk    # Carga ttk (para widgets nuevos 8.5+)
from tkinter import messagebox
import pandas as pd
import operator
import csv

#Declaración de la variable que genera la ventana del programa
raiz = Tk()

def analizar():
    #Declaración de la ventana para visualizar la tabla de
    resultados
    newWindows=Tk()
    #Dar titulo a la nueva ventana
    newWindows.title("Tabla de Similitud")
    #Proceso de lectura de la base de casos que es el archivo csv.
    df = pd.read_csv('winequality-red-Copy1.csv')
    #Guardado de los datos del csv en una lista de lista
    vinos = [list(row) for row in df.values]
    #Declaración de un diccionario para guardar los datos de la
    similitudes
    similares={}

```

```

#Creacion de un arreglo con los valores de los atributos del
caso nuevo para analizar despues,
#que se obtienen de la ventana del programa

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

float(tttsd.get()),float(td.get()),float(tph.get()),float(ts.get())
,float(ta.get()))]
#Declaración de un arreglo con los valores mínimos de los
atributos.
mini=[4.6,0.12,0,0.9,0.012,1,6,0.99,2.74,0.33,8.4]
#Declaración de un arreglo con los valores máximos de los
atributos.
maxi=[15.9,1.58,1.0,13.9,0.611,72.0,289.0,1.0,4.01,2.0,14.9]
#Declaración de un arreglo con los valores de los pesos de
cada atributo del caso nuevo pra analizar despues,
#que se obtienen de la ventana de la aplicación

weight=[float(ctfa.get()),float(ctva.get()),float(ctca.get()),flea
t(ctrs.get()),float(ctc.get()),float(ctfsd.get()),

float(cttsd.get()),float(ctd.get()),float(ctph.get()),float(cts.ge
t()),float(cta.get())]

#Método para obtener la similitud de cada uno de los vinos con
el nuevo.
#Recibira como parametro una lista que contiene los valores de
un vino.
def similarity(ce):
    #Variable para guardar el valor de la similitud.
    valor=0
    #Bucle para calcular la similitud con la formula antes
vista.
    for i in range(len(mini)):
        valor+= weight[i] * (1-((abs(ce[i]-cn[i]))/(maxi[i]-
mini[i])))
    return valor/sum(weight)

#Bucle para recorrer la lista con los vinos y de esta forma
enviar uno por uno al metodo similarity
#para su posterior calculo.
for i in range(len(vinos)):
    #Declaracion de una variable de tipo lista para guarda la
lista que se obtiene de la lista de vinos.
    fila=[]
    #Le asigno un lista que es un vino de la lista de vinos
    fila=vinos[i]
    #Calculo de la similitud para ese vino y guardo en la
variable x
    x = similarity(fila)
    #Aquí uso mi diccionario creado anteriormente para añadir
las similitudes

```

```

similares.update({str(i):round(x,3)})

#Aqui ordenamos los datos de nuestro arreglo de menor a mayor
para luego visualizar en
#la ventana con la tabla.
ordenados = dict(sorted(similares.items(),
key=operator.itemgetter(1)))
#Este arreglo tiene los nombres de las columnas para imprimir
en la tabla.
cols=("#Wine", "Fixed Acidity", "Volatile Acidity", "Citric
Acid","Residual Sugar","Chlorides","Free Sulfure Dioxide","Total
Sulfure
Dioxide","Density","pH","Sulphates","Alcohol","Quality","Similarit
y")
#Aqui declaramos este componente que nos permitira visualizar
en una tabla los datos.
tree = ttk.Treeview(newWindows,columns=cols,show='headings')
#Aqui le otorgamos un panel de navegacion vertical para
moverse por la ventana.
vsb = ttk.Scrollbar(newWindows, orient="vertical",
command=tree.yview)
#Aqui asignamos nuestro panel de navegacion a la ventana.
vsb.pack(side=RIGHT, fill=BOTH)

#Aqui le damos el panel de navegacion a la tabla
tree.configure(yscrollcommand=vsb.set)
#este for recorre mi arreglo de titulos de columnas
for i in range(len(cols)):
    #Asignar a cada columna de la tabla el nombre respectivo
    tree.heading(cols[i],text=cols[i])
    #dar tamaño especifico a cada columna de la tabla
    tree.column(cols[i], minwidth=0, width=50)

#Aqui estamos haciendo que la tabla creada se expanda en la
ventana donde se visualizara.
tree.pack(expand=YES, fill=BOTH)
#Obtengo el tamaño de mi diccionario de resultados.
tam=len(ordenados)
#Bucle para recorrer mi diccionario
for i in range(tam):
    #Con esto obtengo los valores del diccionario ordenados y
mi lista de vinos
    #para darle estos valores a mi tabla en la ventana
    pos=int(list(ordenados.items())[i][0])
    c1=vinos[int(pos)][0]
    c2=vinos[int(pos)][1]
    c3=vinos[int(pos)][2]
    c4=vinos[int(pos)][3]
    c5=vinos[int(pos)][4]
    c6=vinos[int(pos)][5]
    c7=vinos[int(pos)][6]
    c8=vinos[int(pos)][7]
    c9=vinos[int(pos)][8]

```

```

        c10=vinos[int(pos)][9]
        c11=vinos[int(pos)][10]
        c12=vinos[int(pos)][11]
        sim=str(list(ordenados.items())[i][1])
        #Inserto en mi tabla los valores obtenidos para cada fila.

tree.insert("",0,i,values=(str(pos),c1,c2,c3,c4,c5,c6,c7,c8,c9,c10
,c11,c12,sim))

        #Aqui obtengo el vino mas similar y mando a obtener la
posicion
        fpos=list(ordenados.items())[tam-1][0]
        #y el valor de similitud
        fval=list(ordenados.items())[tam-1][1]
        #Aqui obtengo el valor de la similitud para asignar a mi nuevo
vino.
        res=vinos[int(fpos)][11]

        #Aqui creo una lista para pasar a mi archivo csv que es mi
base de datos de vinos

li=[cn[0],cn[1],cn[2],cn[3],cn[4],cn[5],cn[6],cn[7],cn[8],cn[9],cn
[10],res]
        #Aqui pregunto si el vino nuevo ya esta en mi base de datos
        if li in vinos:
            #Si esta solo imprimo los resultados y muestro la tabla
con similitud de uno
            messagebox.showinfo(message="Calificacion: " + res + " ->
Similitud: "+str(fval))
        else:
            #Abrir mi archivo de vinos para escribir el nuevo vino en
caso de no existir.
            with open('winequality-red-Copy1.csv','a') as f:
                #Habilito para escribir sobre el archivo
                writer = csv.writer(f)
                #Asignar el nuevo vino a la base de datos
                writer.writerow((li))
            #Mando a imprimir el resultado y la similitud obtenida
            messagebox.showinfo(message="Calificacion: " + res + " ->
Similitud: "+str(fval))

#Esta lista sirve para signar los valores a los combo box
lista =[0,1,2,3,4,5,6,7,8,9,10]

#Le doy un tamaño a la venta principal
raiz.geometry('600x225') # anchura x altura

#Le doy nombre a la aplicación
raiz.title('Aplicación')

#De aqui en adelante declararemos las etiquetas y los cuadros para
los valores con SpinBox con los

```

```

#valores minimos y maximos y los combo box para los pesos de cada
caracteristica del vino.
Label(raiz,text="ANALISIS DE VINOS ROJOS").place(x=200,y=0)

Label(raiz,text="Fixed Acidy").place(x=0,y=25)
tfa = Spinbox(raiz, from_=4.6, to=15.9 ,
width=5,increment=0.1,font='Helvetica 12')
tfa.place(x=150,y=25)
ctfa = ttk.Combobox(raiz,values=lista,width=5,font='Helvetica 12')
ctfa.place(x=225, y=25)
ctfa.current(3)

Label(raiz,text="Volatily Acidy").place(x= 308,y=25)
tva = Spinbox(raiz, from_=0.12, to=1.58 ,
width=5,increment=0.01,font='Helvetica 12')
tva.place(x=450 ,y=25)
ctva = ttk.Combobox(raiz,values=lista,width=5,font='Helvetica 12')
ctva.place(x=525, y=25)
ctva.current(3)

Label(raiz,text="Citric Acid").place(x=0,y=50)
tca = Spinbox(raiz, from_=0.0, to=1.0 ,
width=5,increment=0.1,font='Helvetica 12')
tca.place(x=150 ,y=50)
ctca = ttk.Combobox(raiz,values=lista,width=5,font='Helvetica 12')
ctca.place(x=225, y=50)
ctca.current(3)

Label(raiz,text="Residual Sugar").place(x= 308,y=50)
trs = Spinbox(raiz, from_=0.9, to=13.9 ,
width=5,increment=0.1,font='Helvetica 12')
trs.place(x=450 ,y=50)
ctrs = ttk.Combobox(raiz,values=lista,width=5,font='Helvetica 12')
ctrs.place(x=525, y=50)
ctrs.current(5)

Label(raiz,text="Chlorides").place(x=0,y=75)
tc = Spinbox(raiz, from_=0.012, to=0.611 ,
width=5,increment=0.001,font='Helvetica 12')
tc.place(x=150 ,y=75)
ctc = ttk.Combobox(raiz,values=lista,width=5,font='Helvetica 12')
ctc.place(x=225, y=75)
ctc.current(1)

Label(raiz,text="Free Sulfur Dioxide").place(x= 308,y=75)
tfstd = Spinbox(raiz, from_=1.0, to=72.0 ,
width=5,increment=1.0,font='Helvetica 12')
tfstd.place(x=450 ,y=75)
ctfstd = ttk.Combobox(raiz,values=lista,width=5,font='Helvetica
12')
ctfstd.place(x=525, y=75)
ctfstd.current(1)

```

```

Label(raiz,text="Total Sulfure Dioxide").place(x=0,y=100)
tttsd = Spinbox(raiz, from_=6.0, to=289.0 ,
width=5,increment=1,font='Helvetica 12')
tttsd.place(x=150 ,y=100)
cttsd = ttk.Combobox(raiz,values=lista,width=5,font='Helvetica
12')
cttsd.place(x=225, y=100)
cttsd.current(1)

Label(raiz,text="Density").place(x= 308,y=100)
td= Spinbox(raiz, from_=0.9900, to=1.0000 ,
width=6,increment=0.0001,font='Helvetica 12')
td.place(x=450 ,y=100)
ctd = ttk.Combobox(raiz,values=lista,width=5,font='Helvetica 12')
ctd.place(x=525, y=100)
ctd.current(1)

Label(raiz,text="pH").place(x=0,y=125)
tph = Spinbox(raiz, from_=2.74, to=4.01 ,
width=5,increment=0.01,font='Helvetica 12')
tph.place(x=150 ,y=125)
ctph = ttk.Combobox(raiz,values=lista,width=5,font='Helvetica 12')
ctph.place(x=225, y=125)
ctph.current(6)

Label(raiz,text="Sulphates").place(x= 308,y=125)
ts= Spinbox(raiz, from_=0.33, to=2.0 ,
width=5,increment=0.01,font='Helvetica 12')
ts.place(x=450 ,y=125)
cts = ttk.Combobox(raiz,values=lista,width=5,font='Helvetica 12')
cts.place(x=525, y=125)
cts.current(1)

Label(raiz,text="Alcohol").place(x=0,y=150)
ta = Spinbox(raiz, from_=8.4, to=14.9 ,
width=5,increment=0.1,font='Helvetica 12')
ta.place(x=150 ,y=150)
cta = ttk.Combobox(raiz,values=lista,width=5,font='Helvetica 12')
cta.place(x=225, y=150)
cta.current(5)

#Declaro el boton para el analisis
ttk.Button(raiz, text='Analizar',
command=analizar).place(x=275,y=200)
#Llamo a mi ventana principal y la visualizo.
raiz.mainloop()

```


Resultados  
Ventana Principal

ANALISIS DE VINOS ROJOS

Fixed Acidy	<input type="text" value="4.6"/>	<input type="text" value="3"/>	Volatily Acidy	<input type="text" value="0.12"/>	<input type="text" value="3"/>
Citric Acid	<input type="text" value="0.0"/>	<input type="text" value="3"/>	Residual Sugar	<input type="text" value="0.9"/>	<input type="text" value="5"/>
Chlorides	<input type="text" value="0.012"/>	<input type="text" value="1"/>	Free Sulfur Dioxide	<input type="text" value="1"/>	<input type="text" value="1"/>
Total Sulfure Dioxide	<input type="text" value="6"/>	<input type="text" value="1"/>	Density	<input type="text" value="0.9900"/>	<input type="text" value="1"/>
pH	<input type="text" value="2.74"/>	<input type="text" value="6"/>	Sulphates	<input type="text" value="0.33"/>	<input type="text" value="1"/>
Alcohol	<input type="text" value="8.4"/>	<input type="text" value="5"/>			

Analizar

Ventana de Resultados  
Cuadro de Calidad del Vino

Calificacion: 6 -> Similitud: 1.0

OK

Ventana de Tabla de Resultado

#Wine	Fixed Acid	Volatile Ac	Citric Acid	Residual Si	Chlorides	Free Sulf	Total Sulf	Density	pH	Sulphates	Alcohol	Quality	Similarity
1599	4.6	0.12	0.0	0.9	0.012	1.0	6.0	0.99	2.74	0.33	8.4	6	1.0
1600	4.8	0.12	0.1	0.9	0.012	1.0	6.0	0.99	2.74	0.33	14.9	6	0.822
1332	8.4	0.39	0.1	1.7	0.075	6.0	25.0	0.9958100	3.09	0.43	9.7	6	0.81
1470	10.0	0.69	0.11	1.4	0.084	8.0	24.0	0.9957799	2.88	0.47	9.7	5	0.809
439	7.0	0.62	0.18	1.5	0.062	7.0	50.0	0.9951	3.08	0.6	9.3	5	0.809
1420	7.8	0.53	0.01	1.6	0.077	3.0	19.0	0.995	3.16	0.46	9.8	5	0.806
1418	7.8	0.53	0.01	1.6	0.077	3.0	19.0	0.995	3.16	0.46	9.8	5	0.806
1392	7.1	0.62	0.06	1.3	0.07	5.0	12.0	0.9942	3.17	0.48	9.8	5	0.806
1338	6.0	0.5	0.0	1.4	0.057	15.0	26.0	0.9944799	3.36	0.45	9.5	5	0.801
1337	6.0	0.5	0.0	1.4	0.057	15.0	26.0	0.9944799	3.36	0.45	9.5	5	0.801
1336	6.0	0.5	0.0	1.4	0.057	15.0	26.0	0.9944799	3.36	0.45	9.5	5	0.801
990	7.7	0.39	0.12	1.7	0.0969999	19.0	27.0	0.9959600	3.16	0.49	9.4	5	0.801
988	7.7	0.39	0.12	1.7	0.0969999	19.0	27.0	0.9959600	3.16	0.49	9.4	5	0.801
143	6.3	0.39	0.08	1.7	0.066	3.0	20.0	0.9954	3.34	0.58	9.4	5	0.799
1085	6.8	0.48	0.08	1.8	0.0740000	40.0	64.0	0.99529	3.12	0.49	9.6	5	0.796
1389	6.7	0.48	0.02	2.2	0.08	36.0	111.0	0.99524	3.1	0.53	9.7	5	0.794
373	7.4	0.55	0.22	2.2	0.106	12.0	72.0	0.9959	3.05	0.63	9.2	5	0.794
161	7.6	0.68	0.02	1.3	0.0720000	9.0	20.0	0.9965	3.17	1.08	9.2	4	0.794
25	6.3	0.39	0.16	1.4	0.08	11.0	23.0	0.9955	3.34	0.56	9.3	5	0.793
1226	7.5	0.58	0.03	4.1	0.08	27.0	46.0	0.99592	3.02	0.47	9.2	5	0.791
1236	7.8	0.55	0.0	1.7	0.07	7.0	17.0	0.99659	3.26	0.64	9.4	6	0.789
48	6.4	0.4	0.23	1.6	0.066	5.0	12.0	0.9958	3.34	0.56	9.2	5	0.788
1273	7.5	0.58	0.2	2.0	0.073	34.0	44.0	0.9949399	3.1	0.43	9.3	5	0.787
1240	7.5	0.61	0.2	1.7	0.076	36.0	60.0	0.9949399	3.1	0.4	9.3	5	0.787
1200	7.7	0.57	0.21	1.5	0.069	4.0	9.0	0.9945799	3.16	0.54	9.8	6	0.787
1197	7.7	0.57	0.21	1.5	0.069	4.0	9.0	0.9945799	3.16	0.54	9.8	6	0.787

En la tabla se muestra los datos del análisis de cada uno de los vinos con la similitud ordenada de mayor a menor, es decir, primero el que más se aparece, otorgándole al vino nuevo la calidad del que más se aparece, si el vino ingresado ya existe aparece con una similitud de 1, que significa que ya está en nuestra base de datos que es el archivo csv.



## **Conclusión**

Podemos determinar que el sistema es muy adecuado para calificar la calidad de los vinos porque tomamos en cuenta algunas características de los vinos para su análisis y posterior calificación, de acuerdo con los vecinos más cercanos y la fórmula de calcular la similitud propuesta por wguillen. Además, podemos determinar que el sistema es tan preciso de acuerdo con la cantidad de vinos existentes en la base de datos, entre más vinos tengamos más posibilidad de sacar una calificación más certera.

## **Webgrafía**

- <https://github.com/wguillen/red-wine-quality-cbr/tree/master/presentation>