



NOMBRE:

ALEX BENAVIDEZ

CARRERA:

INGENIERIA EN SISTEMAS

MATERIA:

SISTEMAS EXPERTOS

PROFESOR:

ING DIEGO QUISIS

FECHA:

22/05/2020

1. OBJETIVOS

- Preprocesar los datos del corpus de acuerdo a las sugerencias desarrolladas por wguillen [github].
- Aplicar las técnicas 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.

2. DESARROLLO DE CONTENIDOS

Para la realización de este trabajo se tiene las características de los vinos almacenados en un archivo csv, las cuales utilizaremos para el análisis. Para este trabajo se implantará el método de Wguillen donde se toma en cuenta los distintos aspectos que conforman a los vinos, para así poder obtener los vinos semejantes a otros, y así poder obtener la calidad de los vinos y que tan similar es con otros vinos.

- **FORMULA PROPUESTA**

- Primeramente, se va sumando los valores de cada uno de los vinos
- Seguidamente se va multiplicando el peso cada característica del vino, y el peso.
- Después se realiza la resta entre el valor de la característica del vino que está almacenado en nuestro archivo csv menos las características del nuevo vino y todo esto menos 1. Finalmente se realiza la división entre la resta de los máximos valores de las características menos los valores mínimos de estos.
- A continuación, se obtienen la similitud el cual es igual a la suma de cada una de las características del vino y esto dividimos por la sumatoria de cada peso de nuestro nuevo vino.
- Después se retorna el valor de la similitud de los vinos.
- Seguidamente los casos similares son almacenados en un diccionario, en donde se procede a ordenarlos en el cual se toma en cuenta el cálculo de la similitud de cada uno de los vinos almacenados en este diccionario. A continuación, nos recomendará el vino que más se asemeja al mismo y le asignamos la calidad del vino más semejante al mismo.
- Finalmente se visualiza en la interfaz la similitud y la calidad del vino más cercano y se visualiza los vinos que se asemejan en función de la similitud. Para finalizar se procede a almacenar el caso que se ha ingresado. Si se vuelve a ingresar un vino con las mismas características que el que ya se ha ingresado tendríamos una cercanía

de 1, pero si se ingresa un vino que es solo semejante se tendría una similitud mayor.

- INTERFAZ PROPUESTA

Para la realización de la interfaz se ha utilizado una librería llamada tkinter para poder implementar los componentes a la interfaz. Como se puede visualizar se ha creado los combobox para poder asignar los valores de las características y los valores de los vinos para ser analizados.

APLICACION VINOS

ANALISIS DE VINOS ROJOS

Fixed Acidity	4.6	3	Volatily Acidity	0.12	3
Citric Acid	0.0	3	Residual Sugar	0.9	5
Chlorides	0.012	1	Free Sulfur Dioxide	1	1
Total Sulfure Dioxide	6	1	Density	0.9900	1
pH	2.74	6	Sulphates	0.33	1
Alcohol	8.4	5			

ANALIZAR

Para poder visualizar las diferentes características de los vinos procedemos a cargar el archivo csv que contiene los vinos con sus características, en donde creamos un método que me permita hacer esta función.

TABLA DE SIMILITUDES

#Wine	Fixed Ac	Volatile	Citric Ac	Residua	Chloride	Free Sul	Total Su	Density	pH	Sulphate	Alcohol	Quality	Similarit
1599	4.6	0.12	0.0	0.9	0.012	1.0	6.0	0.99	2.74	0.33	8.4	6	0.947
1332	8.4	0.39	0.1	1.7	0.075	6.0	25.0	0.99581	3.09	0.43	9.7	6	0.853
1470	10.0	0.69	0.11	1.4	0.084	8.0	24.0	0.99577	2.88	0.47	9.7	5	0.844
1420	7.8	0.53	0.01	1.6	0.077	3.0	19.0	0.995	3.16	0.46	9.8	5	0.837
1418	7.8	0.53	0.01	1.6	0.077	3.0	19.0	0.995	3.16	0.46	9.8	5	0.837
990	7.7	0.39	0.12	1.7	0.09699	19.0	27.0	0.99596	3.16	0.49	9.4	5	0.83
988	7.7	0.39	0.12	1.7	0.09699	19.0	27.0	0.99596	3.16	0.49	9.4	5	0.83
1113	8.9	0.24	0.39	1.6	0.07400	3.0	10.0	0.99698	3.12	0.59	9.5	6	0.827
1110	8.3	0.53	0.0	1.4	0.07	6.0	14.0	0.99593	3.25	0.64	10.0	6	0.822
439	7.0	0.62	0.18	1.5	0.062	7.0	50.0	0.9951	3.08	0.6	9.3	5	0.822

- **IMPLEMENTACION DEL CODIGO**

```
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
import os

principal = Tk()

def analizar():
    ventanaRecomdar=Tk()

    ventanaRecomdar.title("TABLA DE SIMILITUDES")
    ventanaRecomdar.config(bg="red")

    archivo = pd.read_csv('winequality-red.csv')

    lstvinos = [list(row) for row in archivo.values]

    vinosSemejantes={}

    casoNuevo=[float(ta.get()),float(tva.get()),float(tca.get()),float(trs.get()),
float(tc.get()),float(tfsd.get()),

float(tttsd.get()),float(ttd.get()),float(tph.get()),float(ts.get()),float(ta.ge
t())]

    vlsMin=[4.6,0.12,0,0.9,0.012,1,6,0.99,2.74,0.33,8.4]

    vlsMax=[15.9,1.58,1.0,13.9,0.611,72.0,289.0,1.0,4.01,2.0,14.9]

    peso=[float(cta.get()),float(ctva.get()),float(ctca.get()),float(ctrs.get()),f
loat(ctc.get()),float(ctfsd.get()),

float(cttsd.get()),float(ctd.get()),float(ctph.get()),float(cts.get()),float(c
ta.get())]

    def similar(casBase):
        valor=0

        for i in range(len(vlsMin)):
            valor+= peso[i] * (1-((abs(casBase[i]-casoNuevo[i]))/(vlsMax[i]-
vlsMin[i]))))

        return valor/sum(peso)
```

```

for i in range(len(lstvinos)):

    fila=[]

    fila=lstvinos[i]

    x = similar(fila)

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


lstOrden= dict(sorted(vinosSemejantes.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(ventanaRecomdar,columns=cols,show='headings')

vsb = ttk.Scrollbar(ventanaRecomdar, 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)

longitud=len(lstOrden)

for i in range(longitud):

    pos=int(list(lstOrden.items())[i][0])

    c1=lstvinos[int(pos)][0]

    c2=lstvinos[int(pos)][1]

    c3=lstvinos[int(pos)][2]

    c4=lstvinos[int(pos)][3]

    c5=lstvinos[int(pos)][4]

    c6=lstvinos[int(pos)][5]

    c7=lstvinos[int(pos)][6]

    c8=lstvinos[int(pos)][7]

    c9=lstvinos[int(pos)][8]

    c10=lstvinos[int(pos)][9]

    c11=lstvinos[int(pos)][10]

    c12=lstvinos[int(pos)][11]

    sim=str(list(lstOrden.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(lstOrden.items())[longitud-1][0]
fval=list(lstOrden.items())[longitud-1][1]
res=lstvinos[int(fpos)][11]

lsti=[casoNuevo[0],casoNuevo[1],casoNuevo[2],casoNuevo[3],casoNuevo[4],casoNuevo[5],casoNuevo[6],casoNuevo[7],casoNuevo[8],casoNuevo[9],casoNuevo[10],res]

if lsti in lstvinos:
    Label(principal,bg="red",text="Calificacion: " + res + "\n"+"Similitud: "+str(fval), font="Courier 20").place(x=260,y=200)
else:
    with open('winequality-red','a') as f:
        writer = csv.writer(f)
        writer.writerow((lsti))

    Label(principal,bg="red",text="Calificacion: " + res + "\n"+"Similitud: "+str(fval), font="Courier 20").place(x=260,y=200)

lista =[0,1,2,3,4,5,6,7,8,9,10]

principal.geometry('700x325') # anchura x altura

principal.title('APLICACION VINOS')

principal.config(bg="red")

Label(principal,bg="red",text="ANALISIS DE VINOS ROJOS").place(x=240,y=0)

Label(principal,bg="red",text="Fixed Acidity",font="Courier 8").place(x=0,y=25)
fa = Spinbox(principal, from_=4.6, to=15.9 , width=5,increment=0.1,font='Courier 12')
fa.place(x=150,y=25)
cfa = ttk.Combobox(principal,values=lista,width=5,font='Courier 12')
cfa.place(x=225, y=25)
cfa.current(3)

Label(principal,bg="red",text="Volatily      Acidity",font="Courier      8").place(x=308,y=25)
tva      =      Spinbox(principal,      from_=0.12,      to=1.58      ,
width=5,increment=0.01,font='Courier 12')

```

```

tva.place(x=450 ,y=25)

ctva = ttk.Combobox(principal,values=lista,width=5,font='Courier 12')

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

ctva.current(3)


Label(principal,bg="red",text="Citric Acid",font="Courier 8").place(x=0,y=50)

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

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

ctca = ttk.Combobox(principal,values=lista,width=5,font='Courier 12')

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

ctca.current(3)


Label(principal,bg="red",text="Residual      Sugar",font="Courier      8").place(x=
308,y=50)

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

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

ctr = ttk.Combobox(principal,values=lista,width=5,font='Courier 12')

ctr.place(x=525, y=50)

ctr.current(5)


Label(principal,bg="red",text="Chlorides",font="Courier 8").place(x=0,y=75)

tc      =      Spinbox(principal,      from_=0.012,      to=0.611      ,
width=5,increment=0.001,font='Courier 12')

tc.place(x=150 ,y=75)

ctc = ttk.Combobox(principal,values=lista,width=5,font='Courier 12')

ctc.place(x=225, y=75)

ctc.current(1)


Label(principal,bg="red",text="Free Sulfur Dioxide",font="Courier 8").place(x=
308,y=75)

tf = Spinbox(principal, from_=1.0, to=72.0 ,
width=5,increment=1.0,font='Courier 12')

tf.place(x=450 ,y=75)

ctf = ttk.Combobox(principal,values=lista,width=5,font='Courier 12')

ctf.place(x=525, y=75)

ctf.current(1)

```

```

Label(principal,bg="red",text="Total      Sulfure      Dioxide",font="Courier
8").place(x=0,y=100)

ttsd      =      Spinbox(principal,      from_=6.0,      to=289.0      ,
width=5,increment=1,font='Courier 12')

ttsd.place(x=150 ,y=100)

cttsd = ttk.Combobox(principal,values=lista,width=5,font='Courier 12')
cttsd.place(x=225, y=100)
cttsd.current(1)


Label(principal,bg="red",text="Density",font="Courier 8").place(x= 308,y=100)

td=      Spinbox(principal,      from_=0.9900,      to=1.0000      ,
width=6,increment=0.0001,font='Courier 12')

td.place(x=450 ,y=100)

ctd = ttk.Combobox(principal,values=lista,width=5,font='Courier 12')
ctd.place(x=525, y=100)
ctd.current(1)


Label(principal,bg="red",text="pH",font="Courier 8").place(x=0,y=125)

tph      =      Spinbox(principal,      from_=2.74,      to=4.01      ,
width=5,increment=0.01,font='Courier 12')

tph.place(x=150 ,y=125)

ctph = ttk.Combobox(principal,values=lista,width=5,font='Courier 12')
ctph.place(x=225, y=125)
ctph.current(6)


Label(principal,bg="red",text="Sulphates",font="Courier 8").place(x= 308,y=125)

ts=      Spinbox(principal,      from_=0.33,      to=2.0      ,
width=5,increment=0.01,font='Courier 12')

ts.place(x=450 ,y=125)

cts = ttk.Combobox(principal,values=lista,width=5,font='Courier 12')
cts.place(x=525, y=125)
cts.current(1)


Label(principal,bg="red",text="Alcohol",font="Courier 8").place(x=0,y=150)

ta = Spinbox(principal, from_=8.4, to=14.9 , width=5,increment=0.1,font='Courier
12')

ta.place(x=150 ,y=150)

cta = ttk.Combobox(principal,values=lista,width=5,font='Courier 12')
cta.place(x=225, y=150)

```



```

cta.current(5)

ttk.Button(principal, text='ANALIZAR', command=analizar).place(x=150,y=200)

raiz.mainloop()

exi=os.path.exists("myc.csv")

if exi:

    df1 = pd.read_csv("myc.csv")

    lista1 = [list(row) for row in df1.values]

    print(lista1)

else:

    with open('myc.csv','a') as f:

        writer = csv.writer(f)

        columnas=("fixed    Acidity",    "volatile    acidity",    "citric
acid","residual    sugar","chlorides","free    sulfure    dioxide","total    sulfure
dioxide","density","pH","sulphates","alcohol","quality")

        writer.writerow(columnas)

        writer.writerow(list(li))

        Label(principal,bg="red",text="Calificacion: " + res + "\n"+"Similitud:
"+str(fval), font="Courier 20").place(x=260,y=200)

        dfu = pd.read_csv("myc.csv")

        lista1 = [list(row) for row in dfu.values]

        print(lista1)

```

3. RESULTADOS

APLICACION VINOS

ANALISIS DE VINOS ROJOS

Fixed Acidy	4.6	3	Volatily Acidy	0.12	3
Citric Acid	0.0	3	Residual Sugar	0.9	5
Chlorides	0.012	1	Free Sulfur Dioxide	1	1
Total Sulfure Dioxide	6	1	Density	0.9900	1
pH	2.74	6	Sulphates	0.33	1
Alcohol	8.4	5			

ANALIZAR

Calificacion: 6
Similitud: 0.947

TABLA DE SIMILITUDES													
#Wine	Fixed Ac	Volatile	Citric Ac	Residua	Chloride	Free Sul	Total Su	Density	pH	Sulphate	Alcohol	Quality	Similarit
1599	4.6	0.12	0.0	0.9	0.012	1.0	6.0	0.99	2.74	0.33	8.4	6	0.947
1332	8.4	0.39	0.1	1.7	0.075	6.0	25.0	0.99581	3.09	0.43	9.7	6	0.853
1470	10.0	0.69	0.11	1.4	0.084	8.0	24.0	0.99577	2.88	0.47	9.7	5	0.844
1420	7.8	0.53	0.01	1.6	0.077	3.0	19.0	0.995	3.16	0.46	9.8	5	0.837
1418	7.8	0.53	0.01	1.6	0.077	3.0	19.0	0.995	3.16	0.46	9.8	5	0.837
990	7.7	0.39	0.12	1.7	0.09699	19.0	27.0	0.99596	3.16	0.49	9.4	5	0.83
988	7.7	0.39	0.12	1.7	0.09699	19.0	27.0	0.99596	3.16	0.49	9.4	5	0.83
1113	8.9	0.24	0.39	1.6	0.07400	3.0	10.0	0.99698	3.12	0.59	9.5	6	0.827
1110	8.3	0.53	0.0	1.4	0.07	6.0	14.0	0.99593	3.25	0.64	10.0	6	0.822
439	7.0	0.62	0.18	1.5	0.062	7.0	50.0	0.9951	3.08	0.6	9.3	5	0.822

4. CONCLUSIONES

- Se ha realizado de manera correcta esta practica utilizando el método de Wguillen
- Se ha realizado cada uno de los requisitos para poder realizar esta práctica de manera correcta.