

Sistemas Expertos - Recomendación de Vinos

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

Desarrollo

Para el desarrollo del presente trabajos porcederemos a realizar en analisis de un archivo csv, el mismo que contiene diferentes características de los vinos. Para el analisis implementaremos la formula de propuesta por Wguillen donde considera los diferentes aspectos de los vinos para realizar una busqueda de los que mas se asemejan al vino que queremos comparar y nos arroja una recomendacion de la calidad y la similaridad. Dentro del algoritmo K-nn el valor de similaridad va desde 0 hasta 1 donde uno es el valor con mayor similaridad.

Formula

La formula implementada para el calculo de la similaridad de los vinos mas cercanos al introducido en la interfaz lo realizaremos de la siguiente manera:

- Se va sumando los valores de las características de cada vino el mismo que se almacena en una variable
- Se multiplica el peso que tiene cada característica del vino, el peso esta considerado entre 0 - 10
- Se realiza una resta entre 1 menos, la diferencia entre el valor de la característica del caso Base o vino que se encuentra en nuestra Base de casos o archivo csv menos la característica de nuestro nuevo caso o vino a comparar, Esta diferencia se divide entre la diferencia de los valores maximos de cada característica menos los valores minimos de las mismas.
- Finalmente se retorna la similaridad la cual es el valor de la sumatoria de las características del vino dividido para la sumatoria de los pesos de nuestro nuevo caso o vino.
- Retornamos el valor de la formula o similaridad de los vinos.

Los casos que son similares los almacenamos en un Diccionario que nos permitara ordenarlos en funcion del calculo de la similaridad de cada uno, lo cual nos permitira recomendar el vino mas parecido al que se ingresas en la interfaz. A nuestro caso o vino ingresado le agregamos la calidad del vino recomendado ya que el mismo sera el mas cercano o el que mejor se asemeje al vino de nuestra Base de casos.

Nuestra al devolver los vinos mas cercanos al ingresado por la interfaz retornamos la similaridad y la calidad del mas cercano y procedemos a presetar aquellos vinos que mas se asemejan en funcion de la similaridad o formula que calculamos. Al realizar la comparacion nuestro sistema almacenar el caso que ingresamos, ya que el mismo ira aprendiendo de los casos y guardandolos en nuestra Base de casos ya que si luego se

vuelve a ingresar un vino parecido tendremos una mayor cercanía de similaridad entre ellos o puede ser el mismo. Si es el mismo vino con las mismas características nuestro sistema devolverá el valor de 1, ya que el mismo contiene todas las características similares ingresadas en la interfaz.

Interfaz

La interfaz se procedió a desarrollar con la librería tkinter, la cual nos permite crear los componentes de nuestra ventana. Dentro de la ventana Procedemos a crear los Spinbox los que nos permitiera establecer los valores de las características y sus valores del vino a analizar. las características y los rangos de sus valores son:

- "Fixed Acidity"	Valor minimo = 4.6	Valor maximo = 15.9
- "Volatile Acidity"	Valor minimo = 0.12	Valor maximo = 1.58
- "Citric Acid"	Valor minimo = 0	Valor maximo = 1.0,
- "Residual Sugar"	Valor minimo = 0.9	Valor maximo = 13.9
- "Chlorides"	Valor minimo = 0.012	Valor maximo = 0.611,
- "Free Sulfure Dioxide"	Valor minimo = 1	Valor maximo = 72.0
- "Total Sulfur Dioxide"	Valor minimo = 6	Valor maximo = 289.0
- "Density"	Valor minimo = 0.99, 1.0,	
- "pH"	Valor minimo = 2.74	Valor maximo = 4.01
- "Sulphates"	Valor minimo = 0.33	Valor maximo = 2.0
- "Alcohol"	Valor minimo = 8.4	Valor maximo = 14.9

Creamos los métodos que nos permitiera cargar el archivo csv que contiene los vinos o casos Base en un Diccionario, el cual nos permitiera acceder a cada uno y sus características para realizar el cálculo de la fórmula de los mismos. Dentro de la interfaz cada vez que se realice una comparación y se recomiende un vino el caso nuevo que se ingresó se irá almacenado en nuestra base de casos ya que nos permitiera recomendar la siguiente vez un vino con más aproximación posible.

Resultado

El sistema que se tiene, permite calcular la similaridad existente entre los casos en la base de casos o csv y el nuevo caso ingresado mediante la interfaz. Gracias a la fórmula se obtiene una similaridad entre 0 y uno, donde se puede asignar el peso de mayor importancia a cada característica del nuevo caso donde el peso será entre 1 y 10. si no se desea que el vino posea esa característica la desviación de la fórmula se reducirá ya que un peso de 0 no tiene importancia en la fórmula.

tk

Fixed Acidity: 4.6 (weight: 3)

volatile acidity: 0.12 (weight: 3)

citric acid: 0.0 (weight: 3)

residual sugar: 0.9 (weight: 5)

chlorides: 0.012 (weight: 1)

free sulfur dioxide: 1.0 (weight: 1)

total sulfur dioxide: 6 (weight: 1)

density: 0.990 (weight: 1)

pH: 2.74 (weight: 6)

sulphates: 0.33 (weight: 1)

alcohol: 8.4 (weight: 5)

PROCESAR

Calificación: 6

Similitud: 1.0

tabla Vinos Recomendados

#Wine	Fixed Ac	Volatile	Citric Ac	Residual	Chloride	Free Sul	Totxt11	Density	pH	Sulphate	Alcohol	Quality	compar:
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
1602	4.7	0.12	0.0	0.9	0.012	1.0	6.0	0.99	2.74	0.33	8.4	6	0.999
1601	5.0	0.12	0.0	0.9	0.012	1.0	6.0	0.99	2.74	0.33	8.4	6	0.996
1603	5.2	0.12	0.0	0.9	0.012	1.0	6.0	0.99	2.74	0.33	8.4	6	0.995
1604	4.9	0.12	0.0	1.1	0.012	1.0	9.0	0.99	2.74	0.33	8.4	6	0.994
1610	5.3	0.12	0.0	1.8	0.012	1.0	17.0	0.99	2.74	0.33	8.4	6	0.981
1608	4.6	0.12	0.0	1.8	0.019	1.0	20.0	0.99	2.74	0.33	9.0	6	0.971
1605	4.6	0.12	0.0	0.9	0.017	1.0	11.0	0.998	2.74	0.33	9.1	6	0.955
1606	4.6	0.12	0.0	0.9	0.017	1.0	11.0	0.998	2.74	0.42	11.5	6	0.891
1607	8.3	0.12	0.0	1.3	0.012	1.0	6.0	0.99	2.74	1.92	10.6	6	0.874

In [59]:

```

1  from tkinter import *
2  from tkinter import ttk
3  from tkinter import messagebox
4  import pandas as pd
5  import operator
6  import csv
7  import os
8
9  master = Tk()
10 def analizar():
11     ventanaVinosRecomendados = Tk()
12     ventanaVinosRecomendados.title("tabla Vinos Recomendados")
13     archivoA = pd.read_csv('winequality-red-Copy1.csv')
14     listaVinos = [list(row) for row in archivoA.values]
15     listaVinosIguales = {}
16
17     nuevoCaso = [float(txt1.get()), float(txt2.get()), float(txt3.get()), float(txt
18                   float(txt6.get()),
19                   float(txt7.get()), float(txt8.get()), float(txt9.get()), float(txt
20 valoresMinimos = [4.6, 0.12, 0, 0.9, 0.012, 1, 6, 0.99, 2.74, 0.33, 8.4]
21 valoresMaximos = [15.9, 1.58, 1.0, 13.9, 0.611, 72.0, 289.0, 1.0, 4.01, 2.0, 14
22 pesoV = [float(txt110.get()), float(txt22.get()), float(txt33.get()), float(txt
23          float(txt66.get()),
24          float(txt77.get()), float(txt88.get()), float(txt99.get()), float(txt1
25
26 def comparar(casoBase):
27     valor = 0
28     for i in range(len(valoresMinimos)):
29         valor += pesoV[i] * (1 - ((abs(casoBase[i] - nuevoCaso[i])) / (valoresM
30     return valor / sum(pesoV)
31
32 for i in range(len(listaVinos)):
33     fila = []
34     fila = listaVinos[i]
35     x = comparar(fila)
36     listaVinosIguales.update({str(i): round(x, 3)})
37
38 listaVinosOrdenados = dict(sorted(listaVinosIguales.items(), key=operator.itemg
39 cols = (
40     "#Wine", "Fixed Acidity", "Volatile Acidity", "Citric Acid", "Residual Suga
41     "Free Sulfure Dioxide",
42     "Totxt11l Sulfure Dioxide", "Density", "pH", "Sulphates", "Alcohol", "Quali
43 tree = ttk.Treeview(ventanaVinosRecomendados, columns=cols, show='headings')
44 vsb = ttk.Scrollbar(ventanaVinosRecomendados, orient="vertical", command=tree.y
45 vsb.pack(side=RIGHT, fill=BOTH)
46
47 tree.configure(yscrollcommand=vsb.set)
48 for i in range(len(cols)):
49     tree.heading(cols[i], text=cols[i])
50     tree.column(cols[i], minwidth=0, width=50)
51 tree.pack(expand=YES, fill=BOTH)
52 txt11m = len(listaVinosOrdenados)
53 for i in range(txt11m):
54     pos = int(list(listaVinosOrdenados.items())[i][0])
55     c1 = listaVinos[int(pos)][0]
56     c2 = listaVinos[int(pos)][1]
57     c3 = listaVinos[int(pos)][2]
58     c4 = listaVinos[int(pos)][3]
59     c5 = listaVinos[int(pos)][4]

```

```

60     c6 = listaVinos[int(pos)][5]
61     c7 = listaVinos[int(pos)][6]
62     c8 = listaVinos[int(pos)][7]
63     c9 = listaVinos[int(pos)][8]
64     c10 = listaVinos[int(pos)][9]
65     c11 = listaVinos[int(pos)][10]
66     c12 = listaVinos[int(pos)][11]
67     sim = str(list(listaVinosOrdenados.items())[i][1])
68     tree.insert("", 0, i, values=(str(pos), c1, c2, c3, c4, c5, c6, c7, c8, c9,
69 fpos = list(listaVinosOrdenados.items())[txt11m - 1][0]
70 fval = list(listaVinosOrdenados.items())[txt11m - 1][1]
71 res = listaVinos[int(fpos)][11]
72
73 li = [nuevoCaso[0], nuevoCaso[1], nuevoCaso[2], nuevoCaso[3], nuevoCaso[4], nue
74     nuevoCaso[7], nuevoCaso[8], nuevoCaso[9], nuevoCaso[10], res]
75 if li in listaVinos:
76     label2.configure(text="Calificacion: " + res)
77     label3.configure(text="Similitud: " + str(fval))
78 else:
79     with open('winequality-red-Copy1.csv', 'a') as f:
80         writer = csv.writer(f)
81         writer.writerow((li))
82     label2.configure(text="Calificacion: " + res)
83     label3.configure(text="Similitud: " + str(fval))
84
85
86 master.geometry('1100x400')
87
88
89 # PRIMERA FILA
90
91 Label(master, text="Fixed Acidy").place(x=0, y=30)
92 # txt1 = Entry(master).place(x=100,y=30)
93 txt1 = Spinbox(master, from_=4.6, to=15.9, increment=0.1, width=10)
94 txt1.place(x=100, y=30)
95 var = IntVar()
96 var.set(3)
97 txt110 = Spinbox(master, from_=1, to=10, textvariable=var, width=3)
98 txt110.place(x=230, y=30)
99
100 Label(master, text="volatile acidity").place(x=350, y=30)
101 txt2 = Spinbox(master, from_=0.12, to=1.58, increment=0.01, width=10)
102 txt2.place(x=460, y=30)
103 var = IntVar()
104 var.set(3)
105 txt22 = Spinbox(master, from_=1, to=10, textvariable=var, width=3)
106 txt22.place(x=600, y=30)
107
108 Label(master, text="citric acid").place(x=720, y=30)
109 # txt3 = Entry(master).place(x=820,y=30)
110 txt3 = Spinbox(master, from_=0.0, to=1.0, increment=0.1, width=10)
111 txt3.place(x=820, y=30)
112 var = IntVar()
113 var.set(3)
114 txt33 = Spinbox(master, from_=1, to=10, textvariable=var, width=3)
115 txt33.place(x=950, y=30)
116
117 # SEGUNDA FILA
118
119 Label(master, text="residual sugar").place(x=0, y=90)
120 # txt4 = Entry(master).place(x=100,y=90)

```

```
121 txt4 = Spinbox(master, from_=0.9, to=13.9, increment=0.1, width=10)
122 txt4.place(x=100, y=90)
123 var = IntVar()
124 var.set(5)
125 txt44 = Spinbox(master, from_=1, to=10, textvariable=var, width=3)
126 txt44.place(x=230, y=90)
127
128 Label(master, text="chlorides").place(x=350, y=90)
129 # txt5 = Entry(master).place(x=460,y=90)
130 txt5 = Spinbox(master, from_=0.012, to=0.611, increment=0.001, width=10)
131 txt5.place(x=460, y=90)
132 txt55 = Spinbox(master, from_=1, to=10, width=3)
133 txt55.place(x=600, y=90)
134
135 Label(master, text="free sulfur dioxide").place(x=720, y=90)
136 # txt6 = Entry(master).place(x=820,y=90)
137 txt6 = Spinbox(master, from_=1.0, to=72.0, increment=0.5, width=10)
138 txt6.place(x=820, y=90)
139 txt66 = Spinbox(master, from_=1, to=10, width=3)
140 txt66.place(x=950, y=90)
141
142 # TERCERA FILA
143
144 Label(master, text="total sulfur dioxide").place(x=0, y=150)
145 # txt7 = Entry(master).place(x=100,y=150)
146 txt7 = Spinbox(master, from_=6.0, to=289.0, increment=1.0, width=10)
147 txt7.place(x=100, y=150)
148 txt77 = Spinbox(master, from_=1, to=10, width=3)
149 txt77.place(x=230, y=150)
150
151 Label(master, text="density").place(x=350, y=150)
152 # txt8 = Entry(master).place(x=460,y=150)
153 txt8 = Spinbox(master, from_=0.99, to=1.0, increment=0.001, width=10)
154 txt8.place(x=460, y=150)
155 txt88 = Spinbox(master, from_=1, to=10, width=3)
156 txt88.place(x=600, y=150)
157
158 Label(master, text="pH").place(x=720, y=150)
159 # txt9 = Entry(master).place(x=820,y=150)
160 txt9 = Spinbox(master, from_=2.74, to=4.01, increment=0.01, width=10)
161 txt9.place(x=820, y=150)
162 var = IntVar()
163 var.set(6)
164 txt99 = Spinbox(master, from_=1, to=10, textvariable=var, width=3)
165 txt99.place(x=950, y=150)
166
167 # CUARTA FILA
168
169 Label(master, text="sulphates").place(x=0, y=210)
170 # txt10 = Entry(master).place(x=100,y=210)
171 txt10 = Spinbox(master, from_=0.33, to=2.0, increment=0.01, width=10)
172 txt10.place(x=100, y=210)
173 txt100 = Spinbox(master, from_=1, to=10, width=3)
174 txt100.place(x=230, y=210)
175
176 Label(master, text="alcohol").place(x=350, y=210)
177 # txt11 = Entry(master).place(x=460,y=210)
178 txt11 = Spinbox(master, from_=8.4, to=14.9, increment=0.1, width=10)
179 txt11.place(x=460, y=210)
180 var = IntVar()
181 var.set(5)
```

```
182 txt111 = Spinbox(master, from_=1, to=10, textvariable=var, width=3)
183 txt111.place(x=600, y=210)
184
185 ttk.Button(master, text='PROCESAR', command=analizar).place(x=850, y=210)
186 # boton1= ttk.Button(master, text="Sortear", command=sortear).place(x=850,y=210)
187
188
189 label2 = ttk.Label(master, text="Calificacion: ", width=50)
190 label2.place(x=10, y=260)
191 label3 = ttk.Label(master, text="Similitud: ", width=50)
192 label3.place(x=10, y=310)
193
194 master.mainloop()
```

In []:

1