

# Universidad Politécnica Salesiana

**Nombre: Fernando Sanchez**

**Materia: Sistemas Expertos**

**Practica:**

## **Caso práctico de implementación de un sistema de razonamiento basado en casos**

### **I. Preprocesar los datos del corpus de acuerdo a las sugerencias desarrolladas por wguillen [github].**

La base de caso original utilizada para el desarrollo del sistema RBC puede ser que se encuentra en: <https://archive.ics.uci.edu/ml/datasets/Wine+Quality> (<https://archive.ics.uci.edu/ml/datasets/Wine+Quality>) .

### **Instalar API pandas para la manipulación y el análisis de datos**

**pip install pandas**

### **Importar la API pandas para procesar los datos del documento excel "winequality-red.csv"**

```
In [1]: 1 import pandas as pd
```

```
In [2]: 1 df = pd.read_csv("winequality-red.csv", sep=';')
2 lista = [list(row) for row in df.values]
3 print(len(lista))
```

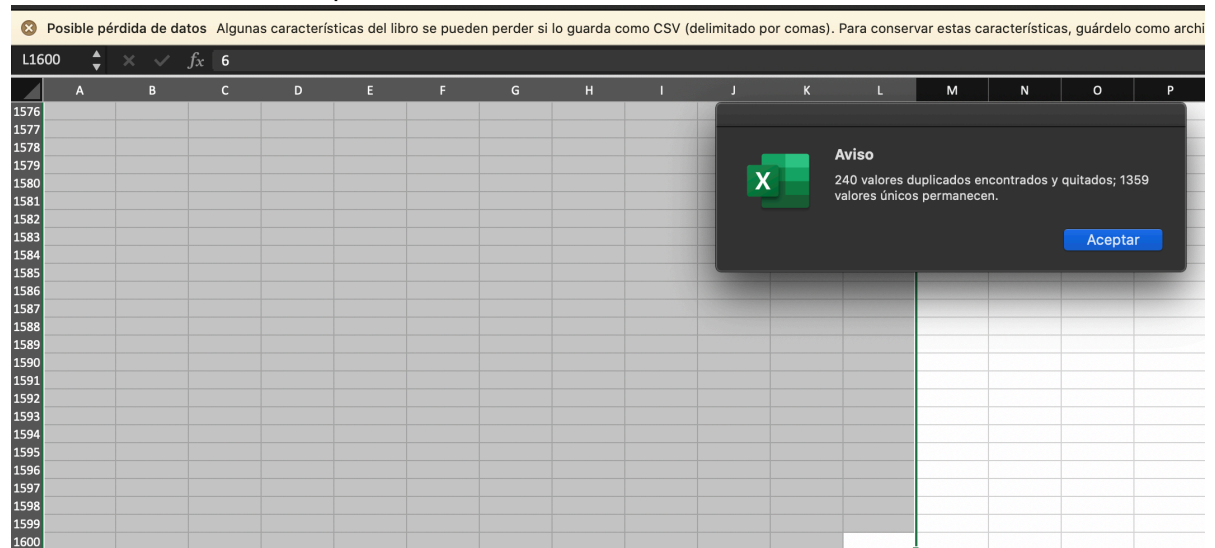
1599

Los casos se filtraron eliminando posibles duplicados y datos erróneos, dejando un total de 1517 cajas en la base utilizada por el sistema.

```
In [3]: 1 result = []
2 for item in lista:
3     if item not in result:
4         result.append(item)
5
6 print(len(result))
```

1359

Verificacion de datos duplicados eliminados



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

El cálculo de similitud entre el caso buscado y los casos de la base de conocimiento ocurre a través de la fórmula:

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

Intervalo de variação!

```
In [4]: 1 val_min = [4.6, 0.12, 0, 0.9, 0.012, 1, 6, 0.99, 2.74, 0.33, 8.
2 val_max = [15.9, 1.58, 1.0, 13.9, 0.611, 72.0, 289.0, 1.0, 4.01
3 def similaridade(ce, cn, weight):
4     valor = 0
5     for i in range(len(val_min)):
6         valor += weight[i] * (1 - (((ce[i] - cn[i])) / (val_max
7     return valor / sum(weight))
```

## Adquirir datos de la interfaz y realizar el calculo de similitud

Se tienen los siguientes atributos del vino (Tabla 2):

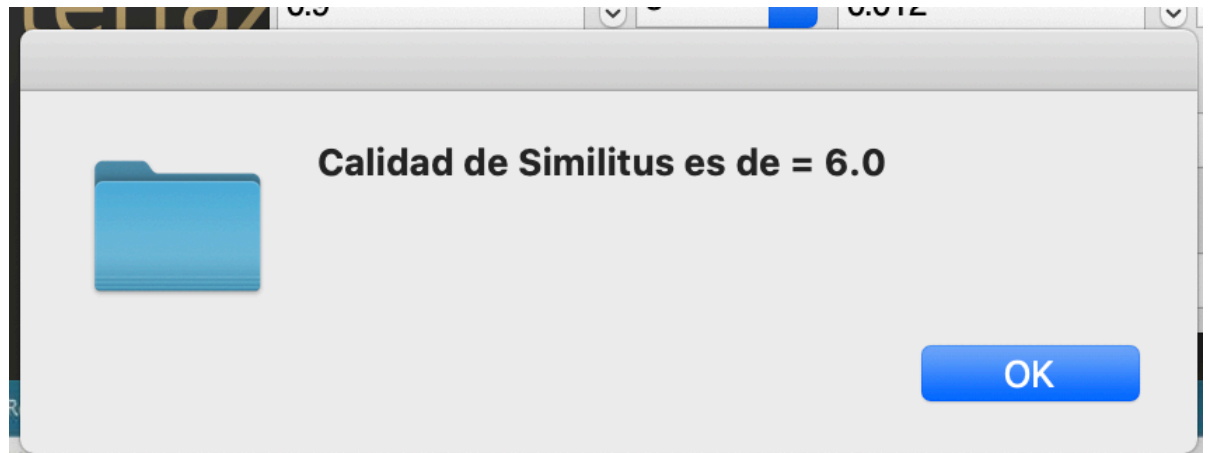
1 - fixed acidity
2 - volatile acidity
3 - citric acid
4 - residual sugar
5 - chlorides
6 - free sulfur dioxide
7 - total sulfur dioxide
8 - density
9 - pH
10 - sulphates
11 - alcohol
Variable de salida:
12 - quality (puntaje entre 0 y 10)

## Crear tabla con los campos establecidos en la imagen anterior

Fixed Acidity	Volatile Acidity	Citric Acid	Residual Sugar	Chlorides	Free Sulfure Dioxide	Total Sulfure Dioxide	Density	pH	Sulphates	Alcohol	Quality	Similarity
0.4	0.25	0.1	1.7	0.075	5.0	25.0	0.9955	3.00	0.12	0.7	5.0	0.91

## Importar messagebox de tkinder

La calidad del vino tinto puntúa entre 0 y 10, como resultado de imprimir en un Mensaje de Información



```
In [44]: 1 from tkinter import messagebox
2 def CalcularSimilitud():
3     tk_Reporte = Tk()
4     tk_Reporte.title("Reporte")
5     tk_Reporte.geometry('1400x600')
6     lista = result
7     similares = {}
8
9     cn = [float(Fixed_Acidty.get()), float(Volatility_Acidty.get()),
10          float(Residual_Sugar.get()), float(Chlorides.get()),
11          float(Total_Sulfure_Dioxide.get()), float(Density.get()),
12          float(Sulphates.get()), float(Alcohol.get())]
13
14     weight = [float(Fixed_Acidty_Peso.get()), float(Volatility_Aci
15               float(Residual_Sugar_Peso.get()), float(Chlorides
16               float(Total_Sulfure_Dioxide_Peso.get()), float(De
17               float(Sulphates_Peso.get()), float(Alcohol_Peso.g
18
19     for i in range(len(lista)):
20         fila = []
21         fila = lista[i]
22         x = similaridade(fila, cn, weight )
23         similares.update({str(i): round(x, 3)})
24
25     ordenados = dict(sorted(similares.items(), key=operator.ite
26     columna = ("Fixed Acidity", "Volatile Acidity", "Citric Aci
27     "Total Sulfure Dioxide", "Density", "pH", "Sulphates", "Alc
28     tree = ttk.Treeview(tk_Reporte, columns=columna, show='head
29     vsh = ttk.Scrollbar(tk_Reporte, command=tree.yview, orient="
```

```

29 vsb.configure(scrollbar=vsb, command=tree.yview, orient=
30 vsb.pack(side=RIGHT, fill=BOTH)
31
32 tree.configure(yscrollcommand=vsb.set)
33 for i in range(len(columna)):
34     tree.heading(columna[i], text=columna[i])
35     tree.column(columna[i], minwidth=0, width=70)
36 tree.pack(expand=YES, fill=BOTH)
37 for i in range(len(ordenados)):
38     pos = int(list(ordenados.items())[i][0])
39     Fixed_Acid_Campo = lista[int(pos)][0]
40     Volatily_Acid_Campo = lista[int(pos)][1]
41     Citric_Acid_Campo = lista[int(pos)][2]
42     Residual_Sugar_Campo = lista[int(pos)][3]
43     Chlorides_Campo = lista[int(pos)][4]
44     Free_Sulfur_Dioxide_Campo = lista[int(pos)][5]
45     Total_Sulfure_Dioxide_Campo = lista[int(pos)][6]
46     Density_Campo = lista[int(pos)][7]
47     pH_Campo = lista[int(pos)][8]
48     Sulphates_Campo = lista[int(pos)][9]
49     Alcohol_Campo = lista[int(pos)][10]
50     Quality_Campo = lista[int(pos)][11]
51     Similary_Campo = str(list(ordenados.items())[i][1])
52     tree.insert("", 0, i, values=( Fixed_Acid_Campo, Volat
53                                     Residual_Sugar_Campo, Chl
54                                     Total_Sulfure_Dioxide_Cam
55                                     Alcohol_Campo, Quality_Ca
56
57
58 aux = list(ordenados.items())[len(ordenados) - 1][0]
59 res = lista[int(aux)][11]
60 messagebox.showinfo(message="Calidad de Similitus es de = ")

```

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

6,7	0,48	0,02	2,2	0,08	36	111	0,995	3,1
7,4	0,55	0,22	2,2	0,106	12	72	0,996	3,05

```
In [ ]: 1 from tkinter import *
2 from tkinter import ttk
3 from tkinter import messagebox
4 import operator
5 tk = Tk()
6
7 lista_Weight = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
8
9 tk.geometry('597x230')
10
11 tk.title('Calidad del Vino Tinto ')
12 tk.resizable(height=FALSE,width=FALSE)
13
14 Label(tk, text="Ingresar Datos").place(x=10, y=0)
15
16 Label(tk, text="Fixed Acidy").place(x=0, y=25)
17 Fixed_Acidty = Spinbox(tk, from_=4.6, to=15.9, width=15, increme
18 Fixed_Acidty.place(x=0, y=50)
19 Fixed_Acidty_Peso = ttk.Combobox(tk, values=lista_Weight, width=
20 Fixed_Acidty_Peso.place(x=130, y=50)
21 Fixed_Acidty_Peso.current(3)
22
23 Label(tk, text="Volatily Acidy").place(x=200, y= 25)
24 Volatily_Acidty = Spinbox(tk, from_=0.12, to=1.58, width=15, inc
25 Volatily_Acidty.place(x=200, y=50)
26 Volatily_Acidty_Peso = ttk.Combobox(tk, values=lista_Weight, wid
27 Volatily_Acidty_Peso.place(x=330, y=50)
28 Volatily_Acidty_Peso.current(3)
29
30 Label(tk, text="Citric Acid").place(x=400, y=25)
31 Citric_Acid = Spinbox(tk, from_=0.0, to=1.0, width=15, incremen
```

```
32 Citric_Acid.place(x=400, y=50)
33 Citric_Acid_Peso = ttk.Combobox(tk, values=lista_Weight, width=
34 Citric_Acid_Peso.place(x=530, y=50)
35 Citric_Acid_Peso.current(3)
36
37 Label(tk, text="Residual Sugar").place(x=0, y=75)
38 Residual_Sugar = Spinbox(tk, from_=0.9, to=13.9, width=15, incr
39 Residual_Sugar.place(x=0, y=100)
40 Residual_Sugar_Peso = ttk.Combobox(tk, values=lista_Weight, wid
41 Residual_Sugar_Peso.place(x=130, y=100)
42 Residual_Sugar_Peso.current(5)
43
44 Label(tk, text="Chlorides").place(x=200, y=75)
45 Chlorides = Spinbox(tk, from_=0.012, to=0.611, width=15, increm
46 Chlorides.place(x=200, y=100)
47 Chlorides_Peso = ttk.Combobox(tk, values=lista_Weight, width=6,
48 Chlorides_Peso.place(x=330, y=100)
49 Chlorides_Peso.current(1)
50
51 Label(tk, text="Free Sulfur Dioxide").place(x=400, y=75)
52 Free_Sulfur_Dioxide = Spinbox(tk, from_=1.0, to=72.0, width=15,
53 Free_Sulfur_Dioxide.place(x=400, y=100)
54 Free_Sulfur_Dioxide_Peso = ttk.Combobox(tk, values=lista_Weight
55 Free_Sulfur_Dioxide_Peso.place(x=530, y=100)
56 Free_Sulfur_Dioxide_Peso.current(1)
57
58 Label(tk, text="Total Sulfure Dioxide").place(x=0, y=125)
59 Total_Sulfure_Dioxide = Spinbox(tk, from_=6.0, to=289.0, width=
60 Total_Sulfure_Dioxide.place(x=0, y=150)
61 Total_Sulfure_Dioxide_Peso = ttk.Combobox(tk, values=lista_Weig
62 Total_Sulfure_Dioxide_Peso.place(x=130, y=150)
63 Total_Sulfure_Dioxide_Peso.current(1)
64
65 Label(tk, text="Density").place(x=200, y=125)
66 Density = Spinbox(tk, from_=0.9900, to=1.0000, width=15, increm
67 Density.place(x=200, y=150)
68 Density_Peso = ttk.Combobox(tk, values=lista_Weight, width=6, f
69 Density_Peso.place(x=330, y=150)
70 Density_Peso.current(1)
71
72 Label(tk, text="pH").place(x=400, y=125)
73 pH = Spinbox(tk, from_=2.74, to=4.01, width=15, increment=0.01,
74 pH.place(x=400, y=150)
75 pH_Peso = ttk.Combobox(tk, values=lista_Weight, width=6, font='
76 pH_Peso.place(x=530, y=150)
77 pH_Peso.current(6)
78
79 Label(tk, text="Sulphates").place(x=0, y=175)
80 Sulphates = Spinbox(tk, from_=0.33, to=2.0, width=15, increment
81 Sulphates.place(x=0, y=200)
82 Sulphates_Peso = ttk.Combobox(tk, values=lista_Weight, width=6,
83 Sulphates_Peso.place(x=130, y=200)
84 Sulphates_Peso.current(1)
85
```



```

86 Label(tk, text="Alcohol").place(x=200, y=175)
87 Alcohol = Spinbox(tk, from_=8.4, to=14.9, width=15, increment=0.1)
88 Alcohol.place(x=200, y=200)
89 Alcohol_Peso = ttk.Combobox(tk, values=lista_Weight, width=6, font=font)
90 Alcohol_Peso.place(x=330, y=200)
91 Alcohol_Peso.current(5)
92
93 ttk.Button(tk, text='Calcular Similitud', command=CalcularSimilitud).place(x=600, y=200)
94 tk.mainloop()

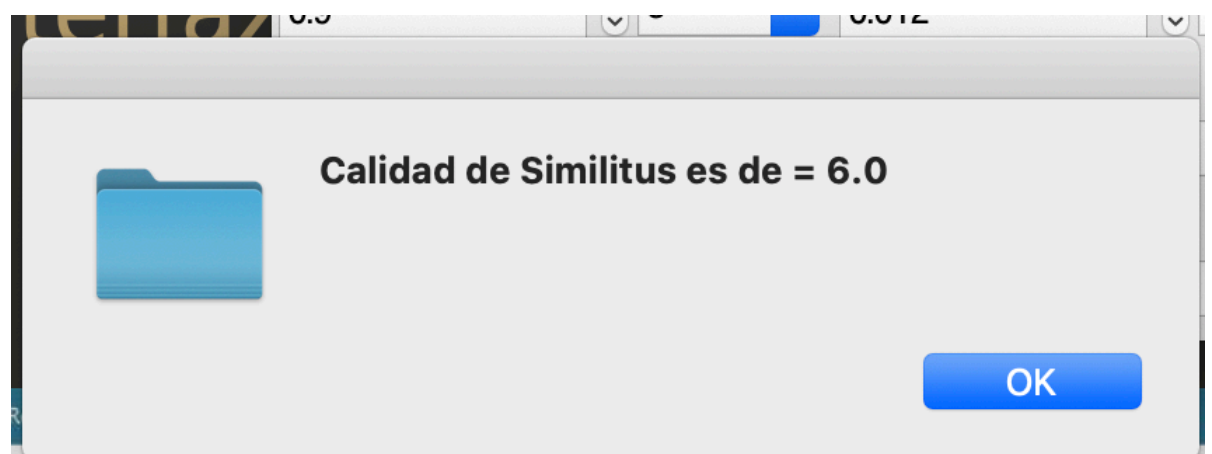
```

## Resultado Obtenidos

Ingresar datos del vino tinto para ver su similitud con datos almacenados en la base de conocimiento:

6,7	0,48	0,02	2,2	0,08	36	111	0,995	3,1
7,4	0,55	0,22	2,2	0,106	12	72	0,996	3,05
6,3	0,30	0,16	1,4	0,09	11	33	0,995	3,24

Mensaje sobre la calidad de similitud del vino tinto ingresado:



Imprimir la tabla de resultado con referencia a la mayor similitud de los datos almacenados:



Reporte												
Fixed Acidity	Volatile Acidity	Citric Acid	Residual Sugar	Chlorides	Free Sulfure Dic	Total Sulfure Di	Density	pH	Sulphates	Alcohol	Quality	Similarity
8.4	0.39	0.1	1.7	0.075	6.0	25.0	0.9958100	3.09	0.43	9.7	6.0	0.81
10.0	0.69	0.11	1.4	0.084	8.0	24.0	0.9957799	2.88	0.47	9.7	5.0	0.809
7.0	0.62	0.18	1.5	0.062	7.0	50.0	0.9951	3.08	0.6	9.3	5.0	0.809
7.8	0.53	0.01	1.6	0.077	3.0	19.0	0.995	3.16	0.46	9.8	5.0	0.806
7.1	0.62	0.06	1.3	0.07	5.0	12.0	0.9942	3.17	0.48	9.8	5.0	0.806
6.0	0.5	0.0	1.4	0.057	15.0	26.0	0.9944799	3.36	0.45	9.5	5.0	0.801
7.7	0.39	0.12	1.7	0.0969999	19.0	27.0	0.9959600	3.16	0.49	9.4	5.0	0.801
6.3	0.39	0.08	1.7	0.066	3.0	20.0	0.9954	3.34	0.58	9.4	5.0	0.799
6.8	0.48	0.08	1.8	0.0740000	40.0	64.0	0.99529	3.12	0.49	9.6	5.0	0.796
6.7	0.48	0.02	2.2	0.08	36.0	111.0	0.99524	3.1	0.53	9.7	5.0	0.794
7.4	0.55	0.22	2.2	0.106	12.0	72.0	0.9959	3.05	0.63	9.2	5.0	0.794
7.6	0.68	0.02	1.3	0.0720000	9.0	20.0	0.9965	3.17	1.08	9.2	4.0	0.794
6.3	0.39	0.16	1.4	0.08	11.0	23.0	0.9955	3.34	0.56	9.3	5.0	0.793
7.5	0.58	0.03	4.1	0.08	27.0	46.0	0.99592	3.02	0.47	9.2	5.0	0.791
7.8	0.55	0.0	1.7	0.07	7.0	17.0	0.99659	3.26	0.64	9.4	6.0	0.789
6.4	0.4	0.23	1.6	0.066	5.0	12.0	0.9958	3.34	0.56	9.2	5.0	0.788
7.5	0.58	0.2	2.0	0.073	34.0	44.0	0.9949399	3.1	0.43	9.3	5.0	0.787
7.5	0.61	0.2	1.7	0.076	36.0	60.0	0.9949399	3.1	0.4	9.3	5.0	0.787
7.7	0.57	0.21	1.5	0.069	4.0	9.0	0.9945799	3.16	0.54	9.8	6.0	0.787
7.1	0.56	0.14	1.6	0.078	7.0	18.0	0.99592	3.27	0.62	9.3	5.0	0.785
8.9	0.24	0.39	1.6	0.0740000	3.0	10.0	0.99698	3.12	0.59	9.5	6.0	0.784
5.2	0.32	0.25	1.8	0.1030000	13.0	50.0	0.9957	3.38	0.55	9.2	5.0	0.784
7.5	0.59	0.22	1.8	0.0819999	43.0	60.0	0.9949899	3.1	0.42	9.2	5.0	0.783

## Conclusiones

Se llego a la conclusión de que es sistemas experto es preciso siempre y cuando se tenga una gran base de conocimiento, como en este caso se tiene 1359 datos con lo que ayudo a determinar la calidad del vini tinto, también se recomienda a dar mantenimiento a la base de conocimiento para no tener datos duplicados o ingresar más datos.

## Referencias

- [1] <https://www.sciencedirect.com/science/article/abs/pii/S0167923609001377?via=ihub>  
(<https://www.sciencedirect.com/science/article/abs/pii/S0167923609001377?via=ihub>)
- [2] <https://github.com/wguilen/red-wine-quality-cbr/tree/master/presentation>  
(<https://github.com/wguilen/red-wine-quality-cbr/tree/master/presentation>)
- [3] <https://archive.ics.uci.edu/ml/datasets/Wine+Quality>  
(<https://archive.ics.uci.edu/ml/datasets/Wine+Quality>)
- [4] <https://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/>  
(<https://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/>)

In [ ]:

1