



Calidad de Vinos

Materia: Sistemas Expertos

Docente: Ing. Diego Quisi

Estudiante: Roberto Pacho

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

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**, compilado por [3].

El corpus se compone de un total de 1599 muestras de vino rojo que contienen información de pruebas fisicoquímicas realizadas en vinos rojos [3].

La información detallada del corpus y las técnicas que aplicaron los autores se puede encontrar en el siguiente enlace: [Artículo](#).

Implementar un sistema CBR básico para determinar la calidad del vino rojo.

Para detectar la similitud usamos la fórmula de Jaccard Similarity

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

- Cálculo del método. El medo lee los datos brindados de winequality-red.scv y calcula la similitud de acuerdo a la forma y a los datos de parametros, que son el nuevo vieno.

<https://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/>

```
In [ ]: from flask import Flask, render_template, request
app = Flask(__name__)

def calcular_similitud_jaccard(valores_vino):
    """
    Metodo que recoree el data set y calcula la similitud con el nuevo vin
    """
    def calculo_similitud(calidad, set_valores_vino):
        """
        Reliza el calculo de similitud de jaccard
        """
        return {
            'calidad': calidad,
            'similitud': len(set_valores_vino.intersection(valores_vino)) /
                len(set_valores_vino.union(valores_vino))
        }

    lista_calculada = []
    with open("/home/RobertoPacho/Documentos/Sistemas_Expertos/winequality
    valores_vino_data = f.readline()
    while valores_vino_data:
        valores_vino_data = f.readline()
        propiedades_vino = valores_vino_data.split(";")
        calidad = propiedades_vino[-1:][0].replace("\n", "")
        lista_calculada.append(calculo_similitud(calidad, set(map(floa
    # Retorna las similitudes y su calidad
    return sorted(lista_calculada, key=lambda item: item['similitud'], rev

@app.route('/calculo_similitud', methods=['POST'])
def calcular_calidad():
    valores_vino = list(map(float, request.form.values()))
    return render_template("index.html", valores=valores_vino, lista=calcu

@app.route('/')
def hello_world():
    return render_template("index.html")

if __name__ == '__main__':
    app.run()
```

- Interfaz usando Flask
- Template Index

The screenshot shows a web browser window with the address bar displaying `127.0.0.1:8000`. The page is titled "Calidad del Vino" and contains two main sections: "DATOS NUEVO VINO" and "TABLA DE SIMILITUDES".

DATOS NUEVO VINO

This section contains a form with the following fields:

- Fixed Acidity
- Volatile Acidity
- Citric Acid
- Residual Sugar
- Chlorides
- Free Sulfur Dioxide
- Volatile Acidity
- Density
- PH
- Sulphates
- Alcohol
- CALCULAR

TABLA DE SIMILITUDES

This section contains a table with the following columns:

Item	Vino Calidad	Vino Similitud
------	--------------	----------------

- Hacemos el ingreso de un nuevo vino y vemos su similitud

The screenshot shows the same web application interface as the previous one, but with the "DATOS NUEVO VINO" form filled out and the "TABLA DE SIMILITUDES" table populated with results.

DATOS NUEVO VINO

The form fields are filled with the following values:

- Fixed Acidity: 5.0
- Volatile Acidity: 6.0
- Citric Acid: 8.0
- Residual Sugar: 3.0
- Chlorides: 4.0
- Free Sulfur Dioxide: 4.0
- Volatile Acidity: 1.0
- Density: 4.0
- PH: 3.0
- Sulphates: 4.0
- Alcohol: 5.0

TABLA DE SIMILITUDES

The table is populated with 15 rows of data:

Item	Vino Calidad	Vino Similitud
1	5	0.14285714285714285
2	6	0.14285714285714285
3	5	0.13333333333333333
4	7	0.13333333333333333
5	4	0.13333333333333333
6	6	0.13333333333333333
7	6	0.13333333333333333
8	5	0.13333333333333333
9	6	0.13333333333333333
10	6	0.13333333333333333
11	6	0.13333333333333333
12	6	0.13333333333333333
13	7	0.13333333333333333
14	5	0.13333333333333333
15	5	0.13333333333333333