

Simulación - Lógica Difusa

Denis Gómez Cruz

C-412

Requerimientos y ejecución:

Se debe contar con el paquete `matplotlib` para la ejecución del proyecto, para esto solo es necesario correr con `python3` el fichero `main.py`. Los resultados obtenidos mediante la ejecución se encontraran en la carpeta `images`, esta contiene todas las funciones de pertenencia usadas y los resultados obtenidos en el fichero `growth.png`.

Problema

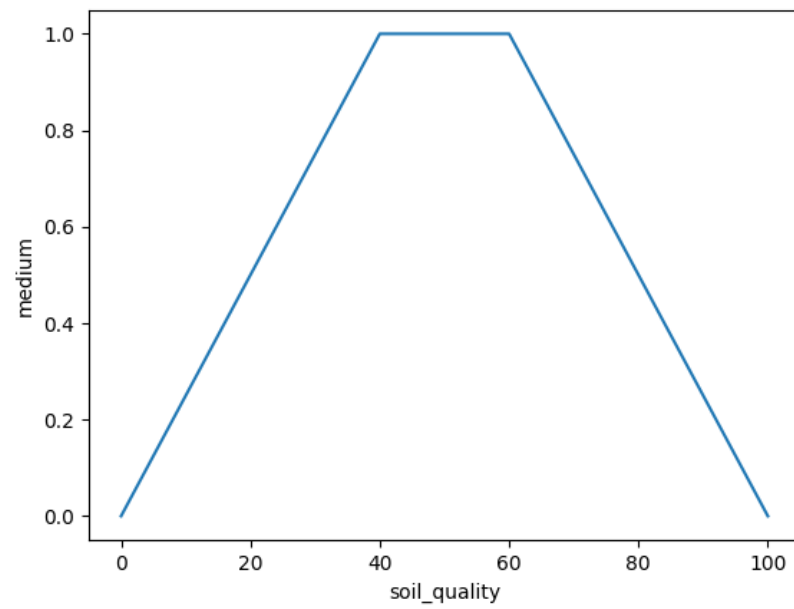
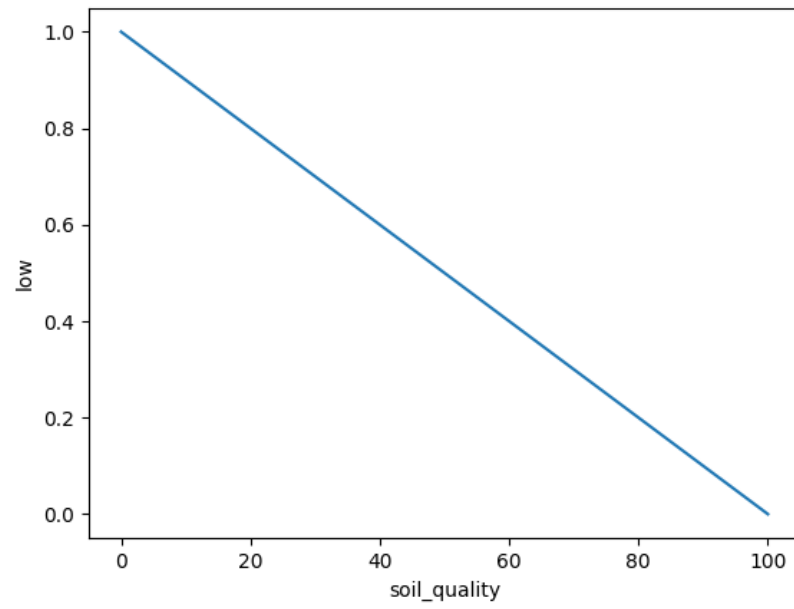
Se quiere analizar la rapidez con que puede crecer una planta bajo ciertas condiciones de algunos de los factores más importantes para el desarrollo de estas como: exposición a luz solar, agua y calidad del suelo. La orientación completa se encuentra en el archivo `orden.pdf`.

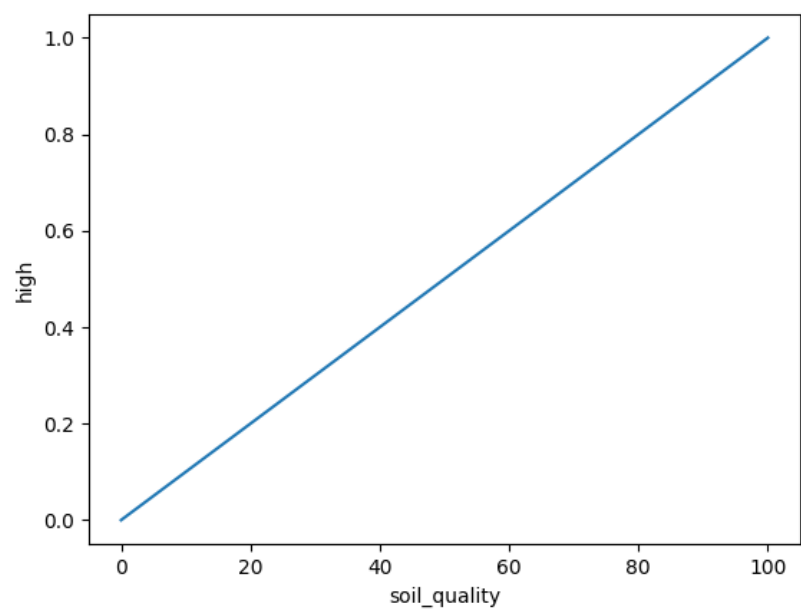
Sistema de Inferencia Difusa

Cada uno de los factores descritos anteriormente tiene tres posibles estados: (bajo, medio, alto)

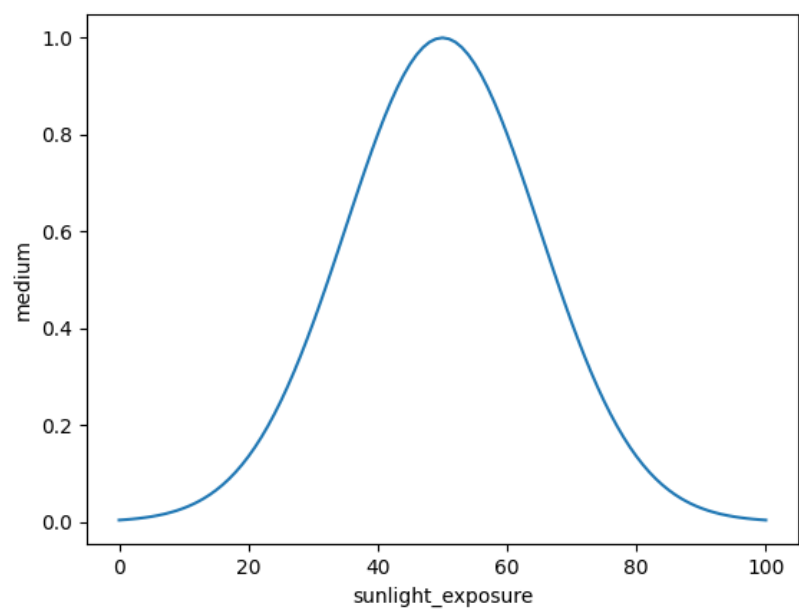
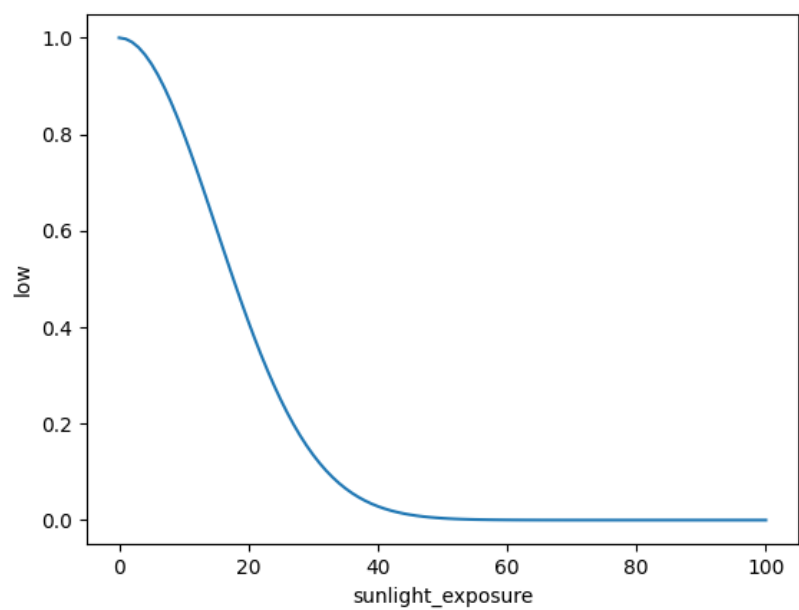
Las funciones de pertenencia en cada caso se muestran a continuación:

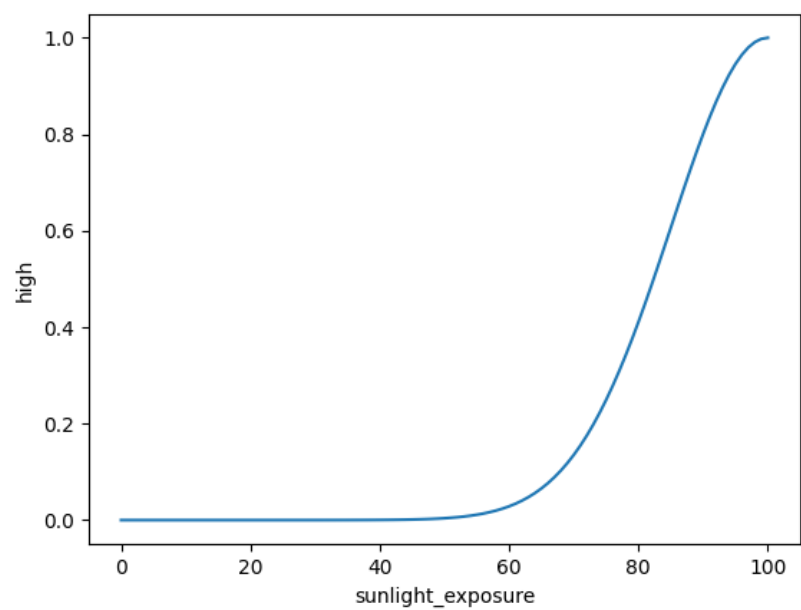
Calidad del suelo:



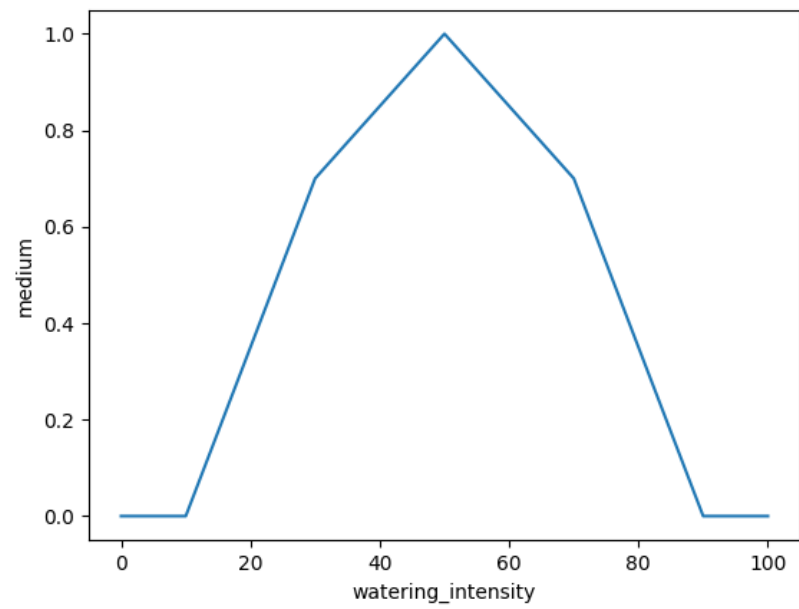
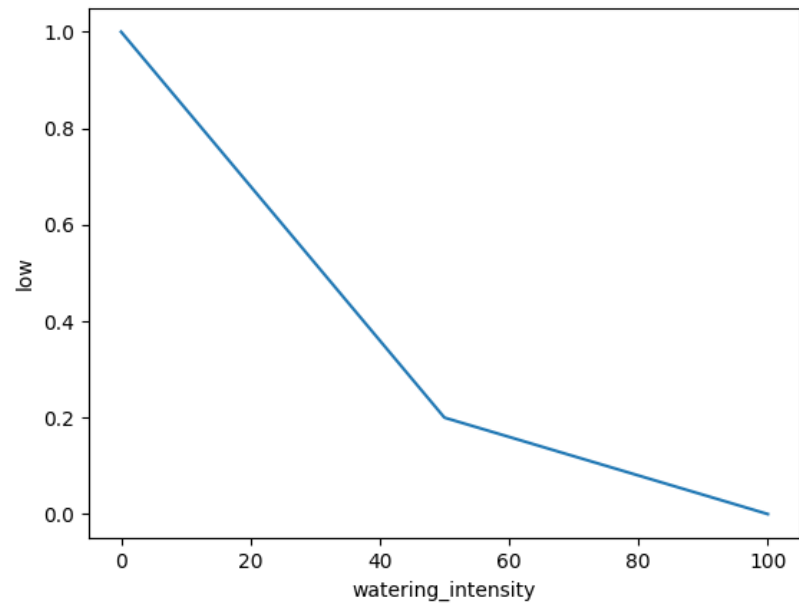


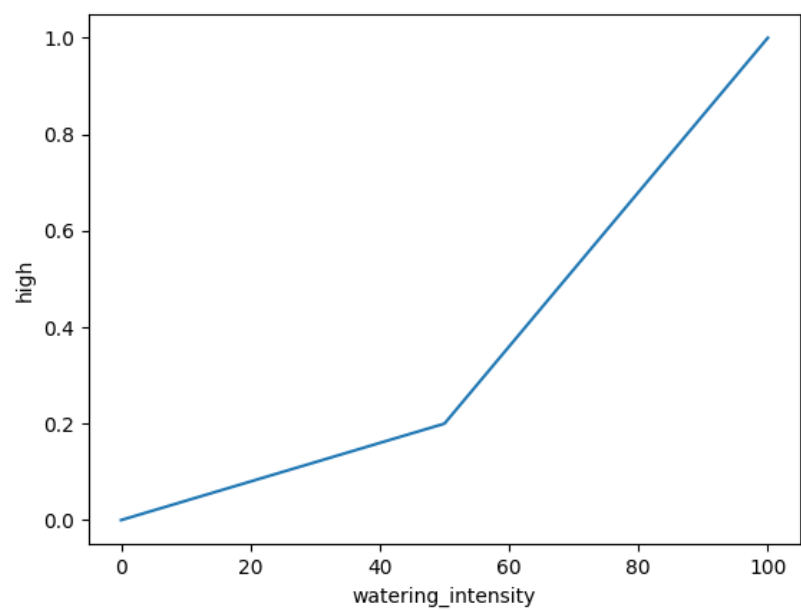
Exposición a la luz solar:



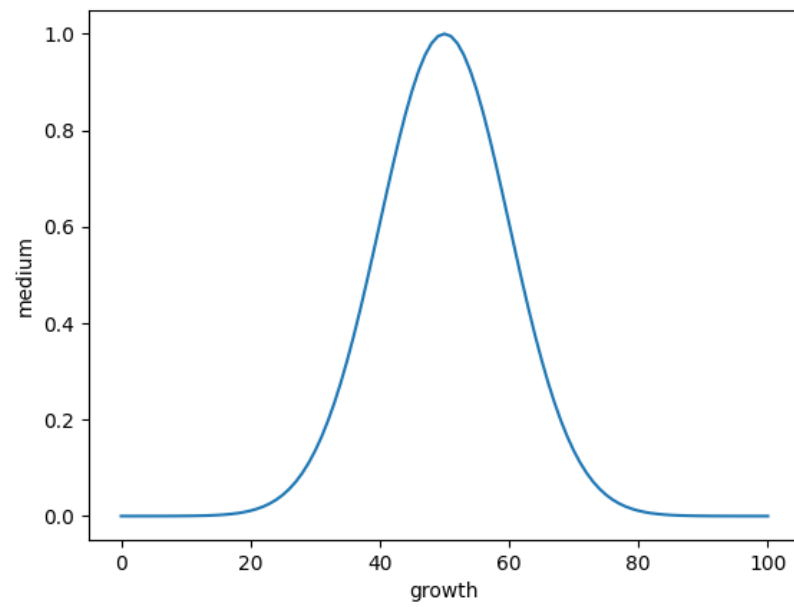
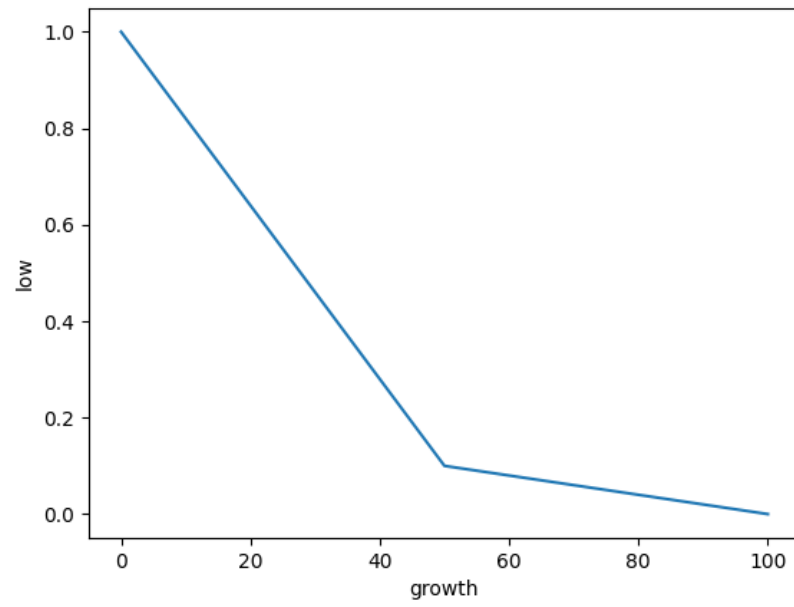


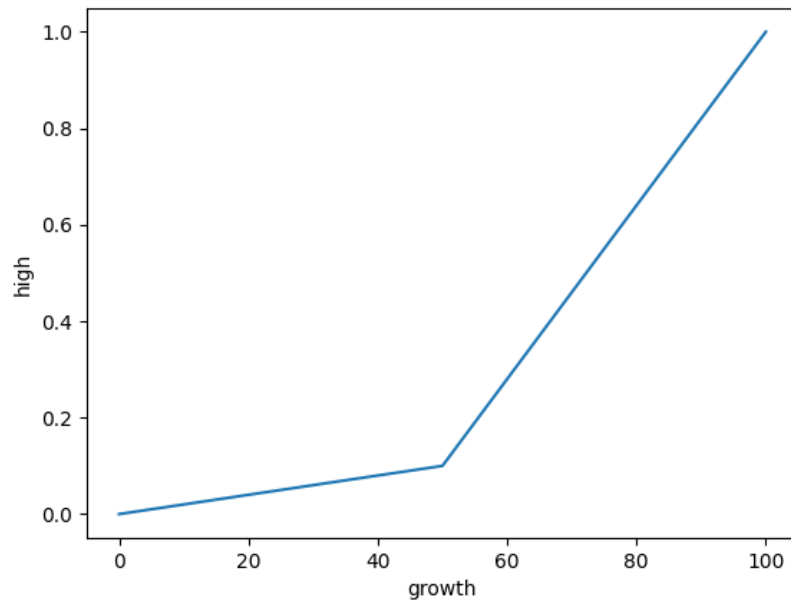
Intensidad del regado:





Nivel de crecimiento:





Como reglas se usó que si al menos dos factores son bajos entonces el crecimiento es bajo, de forma similar para el crecimiento medio y alto. Esto es solo un ejemplo de un posible sistema, las reglas pueden ser modificadas o agregar nuevas reglas de forma simple.

Implementación

Para solucionar el problema anteriormente planteado se creó un sistema de inferencia capaz de modelar predicados difusos, conjuntos difusos, variables lingüísticas y realizar operaciones entre estos como AND, OR, etc... (véase `fuzzy_system.py`)

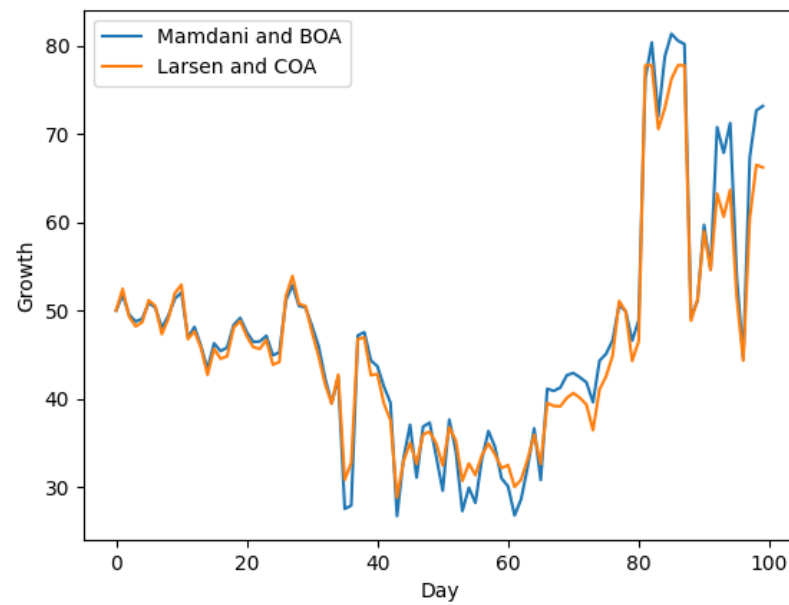
También se usaron funciones de pertenencia como las triangulares, trapezoidales y algunas otras. (véase `fuzzy_number.py`)

Como métodos de inferencia y agregación se implementaron `mamdani` y `larsen` (véase `fuzzy_inference.py`)

Para la defusificación se implementaron los métodos de `bisección`, `máximo` y `centroide` (véase `defuzzification.py`)

Simulación

En el archivo `main.py` se encuentra la simulación del problema, para esta se tomaron los datos de los factores relevantes durante 100 días consecutivos, usando varias distribuciones como la uniforme y gauss para generarlos.



Aquí se pueden apreciar los resultados obtenidos usando como método de inferencia `mamdani` y como método de defusificación `bisección` y de igual forma usando `larsen` y `centroide`. El gráfico muestra el nivel de crecimiento esperado por cada día en una escala del 0 al 100.