Implementación de la Lógica Difusa en un Sistema Experto

Daniel Mejia Naranjo[#1], Manuel Rodriguez[*2], Steven Sanchez [#3]

[#]*Escuela de Ingeniería , Institución Universitaria Salazar y Herrera*
Medellin, Colombia

[1]daniel.mejian@comunidad.iush.edu.co

[3]manuel.rodriguezm@comunidad.iush.edu.co

steven.sanchezr@comunidad.iush.edu.co

***Abstract*— The aim of this paper is to show and explain how an expert system written in python, about an airline, works when you implement Fuzzy Logic by using the "scikitfuzzy" library**

I. Introduccion

La lógica difusa es una extensión de la lógica booleana de Lotfi Zadeh en 1965 basada en la teoría matemática de conjuntos difusos, que es una generalización de la teoría de conjuntos clásica.

Al introducir la noción de grado en la verificación de una condición, permite que una condición esté en un estado distinto a verdadero o falso, la lógica difusa proporciona un valor muy valioso para la flexibilidad para el razonamiento, lo que permite tener en cuenta las imprecisiones e incertidumbres. Para ejemplificar la definición de lógica difusa, desarrollamos a lo largo de este trabajo, un sistema de inferencia difusa cuyo objetivo específico graficar una serie de reglas para aplicar descuentos.

II. Desarrollo Del Contenido

A. Funcionamiento del sistema

El sistema experto se desarrolló con la nueva API de scikit fuzzy (skfuzzy control API). La cual es más simple que la anterior. Este sistema experto es un sistema de tiquetes de aerolínea, Se basa en un problema de "tipping". Este sistema tiene 4 entradas y 4 salidas. Ya sea para descuentos o por ejemplo sugerir el tipo de clase que debería escoger el usuario dependiendo de que comodidad desea y su presupuesto.

Programación Del Sistema

B. Importación de librerías

Se empezó importando las librerías: numpy se abrevió cómo np, skfuzzy cómo fuzz y matplotlib.plot cómo plt; también se importó el módulo control de la librería skfuzz cómo ctrl

C. Antecedentes o Entradas

Luego se crearon las entradas a utilizar

- CantPersonas: Esta entrada Nos va a servir para que el sistemas mas adelante nos sugiera posibles descuentos dependiendo de la cantidad de personas, Se toman como máximo 6 personas para reserva de tiquetes.
- Presupuesto: Esta entrada nos sirve para saber cuál va a ser el presupuesto del usuario en dólares, con un mínimo de 100 y máximo de 999 dólares
- Comodidad: Esta entrada es para saber que tanta comodidad en una escala de 1 al 10 busca el usuario.
- Meses: Esta entrada es para saber que más desea el viaje, esto nos permite saber que tipo de temporada es (Un total de 12 meses)

D. Consecuencias o Salidas

Después se crean las salidas

- Aumento: Esta salida nos arroja un valor de porcentaje para saber cuánto aumentó sugiere el sistema por temporada alta

- Clase: Esta salida dependiendo del presupuesto y comodidad del usuario nos arroja el tipo de clase que sugiere el sistema
- Descuento: Esta salida nos arroja un descuento por la cantidad de personas que van a comprar el tiquete
- SinDescuento: Esta salida nos arroja descuento muy bajos por el tipo de temporada

E. Funciones de Membresía

Ya que se crearon las salidas, continuamos con las funciones de membresía

- Presupuesto: Esta función de pertenencia nos dice que el presupuesto es Poco cuando va de 100 a 300, _demo_ cuando va de 301-600 y Mucho cuando va de 601 - 1000
- Comodidad: Esta función de pertenencia nos dice que la comodidad está en un rango de Poca,Media,Mucha con valores de 1-3 3-6 6-10 respectivamente
- Clase: Esta función de pertenencia nos dice que hay 3 tipos de clases Ejecutiva,Económica y Primera Clase con valores de 1-2-3 respectivamente
- CantPersonas: Esta función de pertenencia se auto completa con .autofm() (Tiene 3 valores: 3,5,7) En este caso utilizamos el valor 5 que nos dice que tienen 5 rangos los cuales son: poor,mediocre,average,decent,good.
- Meses: Esta función de pertenencia es para los meses del 1-12 (Enero,febrero...)
- Aumento: Esta función de pertenencia nos dice que hay un aumento en porcentaje del 0-15%
- SinDescuento: Esta función nos dice que hay un descuento muy pequeño (0-1%)
- Descuento: Esta función de pertenencia nos dice que hay 3 tipos de descuentos bajo medio y alto.

F. Reglas

Por último se programan las reglas:

- Rule 2: Si la cantidad de personas es POOR o MEDIOCRE el descuento es BAJO
- Rule 3: Si la cantidad de personas es DECENT O AVERAGE el descuento es MEDIO
- Rule 4: Si la cantidad de personas es GOOD el descuento es ALTO
- Rule 5: Si los meses son: Enero o Febrero o Noviembre o Diciembre el Aumento es ALTO
- Rule 6: Si los meses son: Marzo o Abril o Mayo o Junio o Julio o Agosto o Septiembre o Octubre no tiene descuento significativo (SinDescuento)
- Rule 7: Si el presupuesto es POCO o la comodidad es POCA la clase sugerida es ECONOMICA
- Rule 8: Si el presupuesto es MEDIO o la comodidad es MEDIA la clase sugerida es EJECUTIVA
- Rule 9: Si el presupuesto es MUCHO o la comodidad es MUCHA la clase sugerida es Primera Clase

G. Sistema de Control y simulación

Creamos a ControlSystemSimulation. Se puede pensar en este objeto como que representa un controlador aplicado a un conjunto específico de circunstancias.

Se dividió en diferentes Partes debido a que dependiendo de la regla tienen diferentes entradas.

H. Aplicación de las reglas, Inputs y procesamiento de números

Reglas 2,3,4

- Descuentos_ctrl = ctrl.ControlSystem([rule2,rule3,rule4])
- Descuentos = ctrl.ControlSystemSimulation(Descuentos_ctrl)

Regla 5

- Aumentos_ctrl = ctrl.ControlSystem([rule5])
- Aumentos = ctrl.ControlSystemSimulation(Aumentos_ctrl)

Regla 6

- NoDescuento_ctrl = ctrl.ControlSystem([rule6])

- NoDescuento = ctrl.ControlSystemSimulation(NoDescuento_ctrl)

Reglas 7,8,9

- Clases_ctrl = ctrl.ControlSystem([rule7,rule8,rule9])
- Clases = ctrl.ControlSystemSimulation(Clases_ctrl)

Inputs

- Descuentos.input['Cantidad de Personas'] = 2
- Aumentos.input['Meses'] = 3
- NoDescuento.input['Meses'] = 3
- Clases.input['Comodidad'] = 8
- Clases.input['Presupuesto'] = 800

Procesamiento de Números

- Descuentos.compute()
- Aumentos.compute()
- NoDescuento.compute()
- Clases.compute()

III. Descripción de Resultados

Este sistema nos arroja 4 resultados los cuales son:

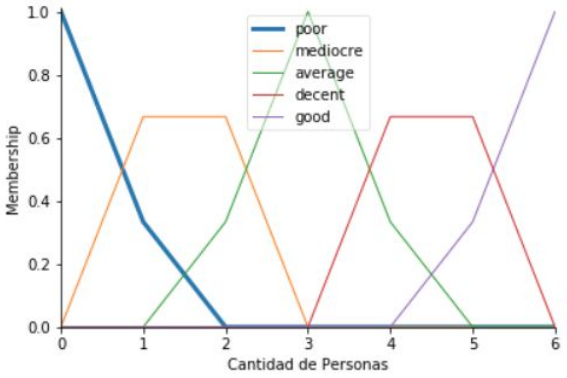
Descuentos: Nos arroja un número el cual es un porcentaje de descuento del valor del tiquete por la cantidad de personas.

Aumentos: Nos arroja un número que es un porcentaje de aumento del valor del tiquete por ser temporada alta.

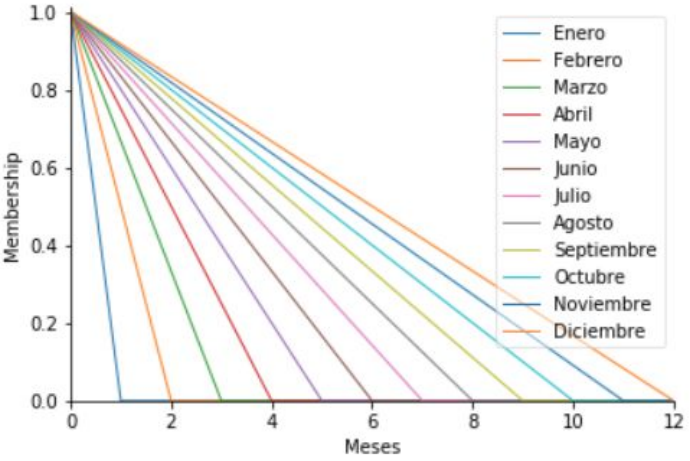
NoDescuento: Nos arroja un número el cual es un porcentaje de descuento muy mínimo del valor del tiquete por ser temporada baja

Clases: Este resultado nos sugiere qué tipo de clase debería escoger el usuario dependiendo de su comodidad y presupuesto (En un rango de 3 clases: Económica, Ejecutiva y Primera Clase

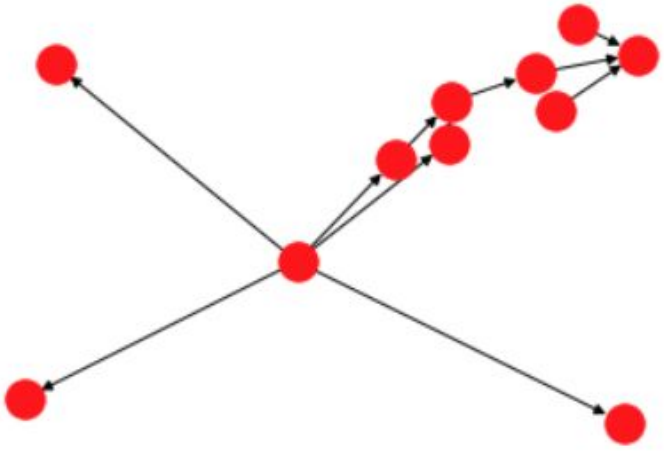
GRAFICA 1
Vista de CantPersonas ['poor']



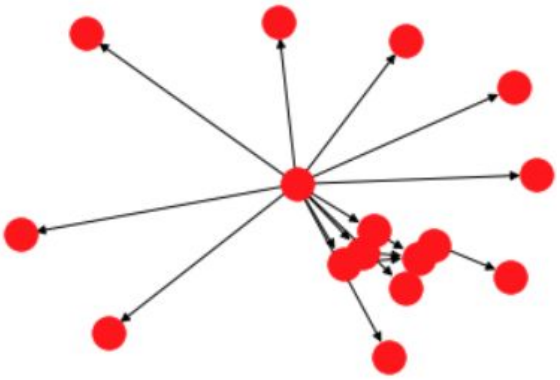
GRAFICA 2
Vistas Meses



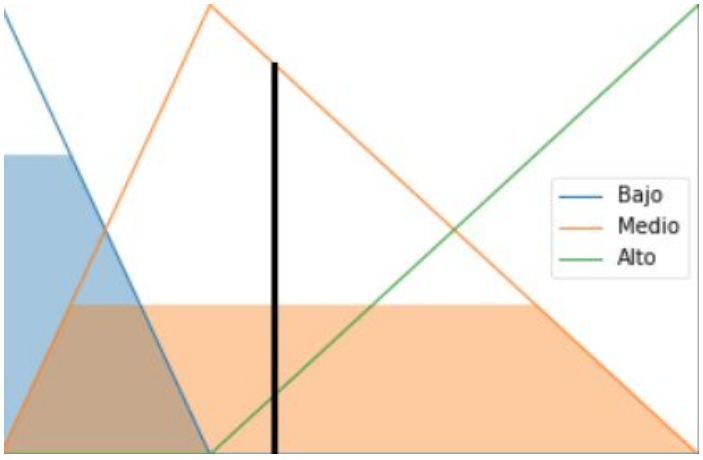
GRAFICA 3
Vistas de la Rule 2



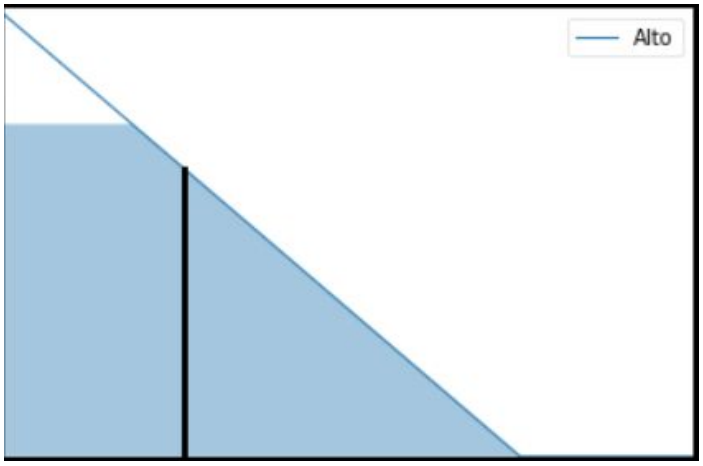
GRAFICA 4
VISTAS DE LA RULE 5



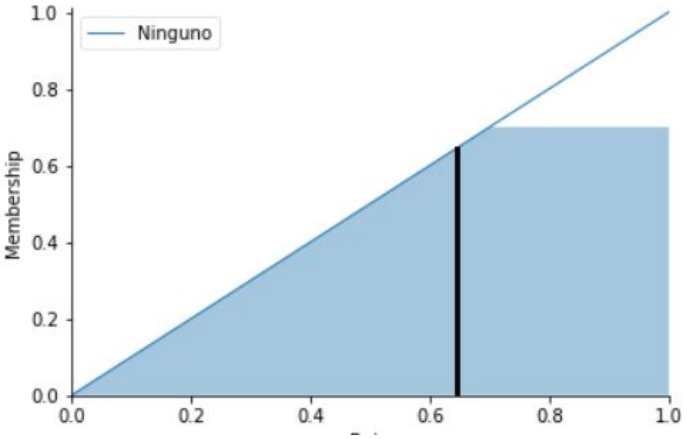
GRAFICA 5
DESCUENTOS



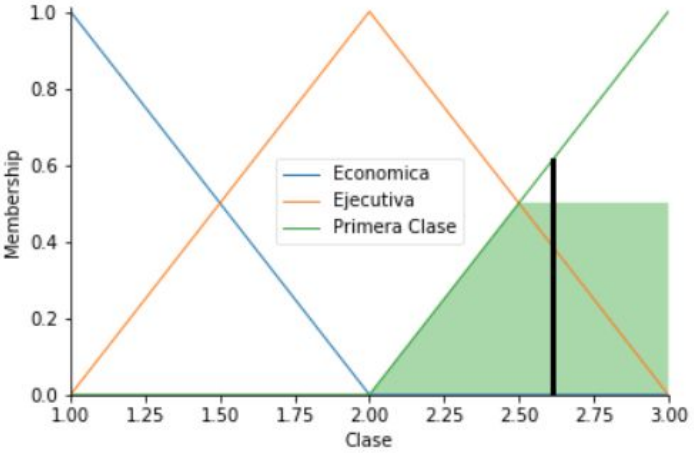
GRAFICA 6
AUMENTOS



GRAFICA 7
SIN DESCUENTO



GRAFICA 8
VISTA DE LAS CLASES



IV. CONCLUSIONES

1. Una ventaja de la lógica difusa para formalizar el razonamiento humano es que las reglas se establecen en un lenguaje natural
2. La librería scikit-fuzzy aumenta el atractivo de Python científico como una alternativa válida a las opciones de código cerrado.
3. La logica difusa nos da una forma alternativa de representar lingüísticamente y subjetivamente atributos del mundo real en la programación.

V. Referencias

[1] Pythonhosted.org. (2017). skfuzzy 0.2 docs — skfuzzy v0.2 docs. [online] Available at: https://pythonhosted.org/scikit-fuzzy/auto_examples/plot_tipping_problem_newapi.html [Accessed 27 Sep. 2018].

[2] Pythonhosted.org. (2017). API Reference — skfuzzy v0.2 docs. [online] Available at: https://pythonhosted.org/scikit-fuzzy/api/api.html [Accessed 27 Sep. 2018].

[3] Biblioteca.udep.edu.pe. (2018). LÓGICA DIFUSA Y SISTEMAS DE CONTROL. [online] Available at: http://www.biblioteca.udep.edu.pe [Accessed 27 Sep. 2018].

[4] Pythonhosted.org. (2017). The Tipping Problem - The Hard Way — skfuzzy v0.2 docs. [online] Available at: https://pythonhosted.org/scikit-fuzzy/auto_examples/plot_tipping_problem.html [Accessed 27 Sep. 2018].