**EXAMEN FINAL DE SISTEMAS EXPERTOS**

**Enunciado:**

Se desea generar un sistema de recomendación de películas, por tal motivo se va a utiliza una base de datos orientada a grafos y un control de lógica difusa para clasificar el riesgo financiero, el mismo que sera ingresado como atributo del cliente en el sistema re comendador, para lograr esto se describe los pasos a seguir:

1) Evaluar el riesgo financiero de sus clientes que requieren la recomendación de películas. Para evaluar el riesgo financiero se toma en cuenta la edad del asegurado y su porcentaje de manejo durante el año. Para ello se tiene las siguientes reglas y la función de pertinencia. El proceso seguir se describe en el siguiente link: [https://medium.com/@javierdiazarca/l%C3%B3gica-difusa-ejercicios-propuestos-b99603ef1bc0](https://medium.com/@javierdiazarca/l%C3%B3gica-difusa-ejercicios-propuestos-b99603ef1bc0" \t "_blank).

2) Generar números aleatorios para la edad y el porcentaje de manejo con el objetivo de generar al menos 100 personas y ademas incluir el listado de películas vistas y el valor del rating de cada película. Al menos 50 películas y un total de nodos de al menos 350 nodos.

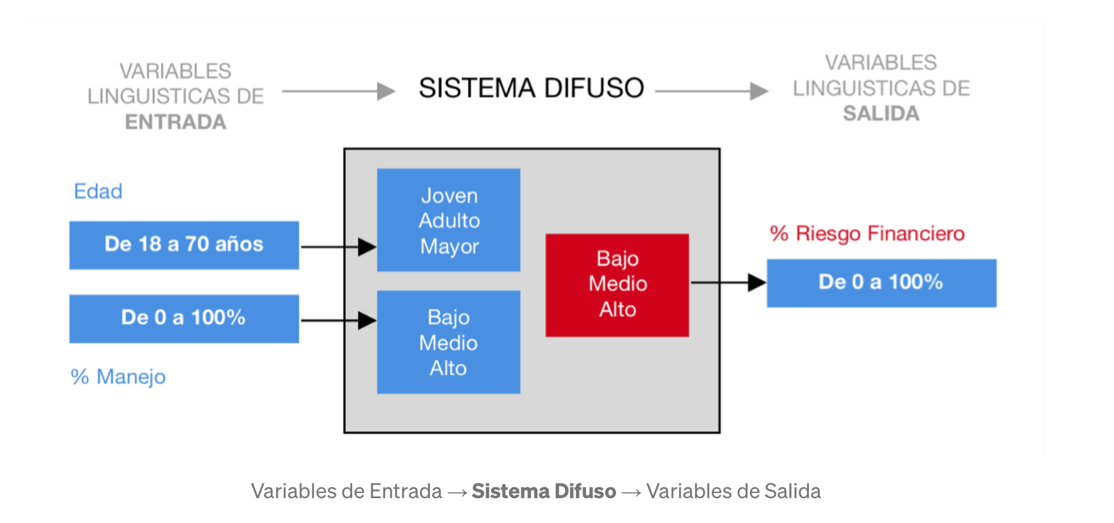
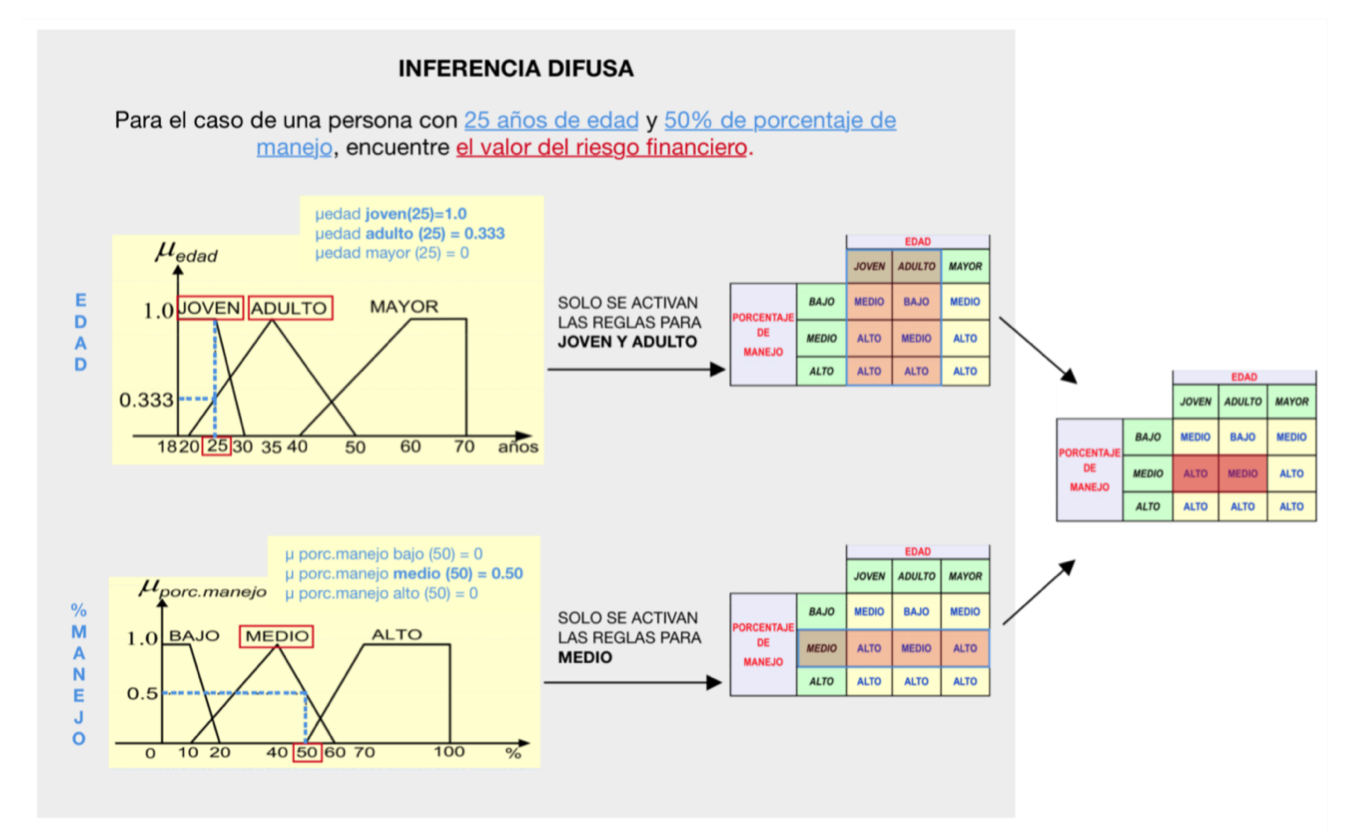
3) Con estos datos aplicar el algoritmo de KNN y Similitud de Coseno para la recomendación de películas, seguir el siguiente tutorial: [https://neo4j.com/graphgist/movie-recommendations-with-k-nearest-neighbors-and-cosine-similarity](https://neo4j.com/graphgist/movie-recommendations-with-k-nearest-neighbors-and-cosine-similarity" \t "_blank).

4) Finalmente realizar alguna interfaz para poder acceder a la recomendación e ingreso de datos y resultados de los procesos.

Generar el Informe en PDF y subir los scripts al repositorio Git para su evaluación. Criterios de Evaluación

* Sistema lógico difuso: 30%
* Neo4J Knn: 30%
* Informe y resultados: 20%
* GUI. programación y pruebas: 20%

**DESARROLLO DEL RIESGO FINANCIERO**

1. **Declaramos las variables**
2. **Obtenemos las reglas**

**CODIGO PARAIMPLEMENTAR LA LOGCA DIFUSA**

**import** numpy **as** np

**import** skfuzzy **as** fuzz

**import** random

**from** skfuzzy **import** control **as** ctrl

​

*# New Antecedent/Consequent objects hold universe variables and membership*

*# functions*

edad **=** ctrl.Antecedent(np.arange(18, 70, 1), 'edad')

manejo **=** ctrl.Antecedent(np.arange(0, 100, 1), 'manejo')

RF **=** ctrl.Consequent(np.arange(0, 100, 1), 'RF')

​edad.automf(3)

manejo.automf(3)

​

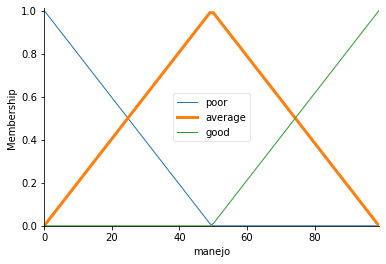
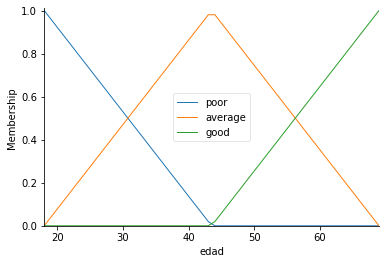
*# Pythonic API*

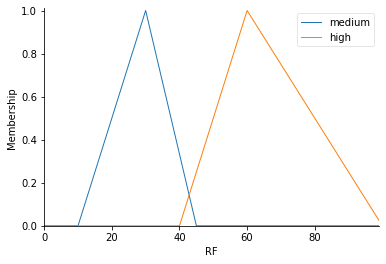
RF['medium'] **=** fuzz.trimf(RF.universe, [10, 30, 45])

RF['high'] **=** fuzz.trimf(RF.universe, [40, 60, 100])

edad.view()

manejo['average'].view()

RF.view()



*#reglas*

rule1 **=** ctrl.Rule(edad['poor'] **|** manejo['average'], RF['high'])

rule2 **=** ctrl.Rule(edad['average'] **|** manejo['average'], RF['medium'])

print(rule1)

print(rule2)

​

IF edad[poor] OR manejo[average] THEN RF[high]

AND aggregation function : fmin

OR aggregation function : fmax

IF edad[average] OR manejo[average] THEN RF[medium]

AND aggregation function : fmin

OR aggregation function : fmax

*#Ahora que tenemos nuestras reglas definidas, podemos simplemente crear un sistema de control a través de:*

tipping\_ctrl **=** ctrl.ControlSystem([rule1, rule2])

tipping **=** ctrl.ControlSystemSimulation(tipping\_ctrl)

Iedad **=** int(random.randrange(18,70))

Imanejo **=** int(random.randrange(0,100))

print(Iedad)

print(Imanejo)

​

*#tipiamos los datos randomicos*

tipping.input['edad'] **=** Iedad

tipping.input['manejo'] **=** Imanejo

tipping.compute()

48

69

*#Una vez calculado, podemos ver el resultado así como visualizarlo.*

​

**if** tipping.output['RF'] **>** 50:

print("su riesgo finaciero es Alto, Peligro: ")

print()

print (tipping.output['RF'])

**else**:

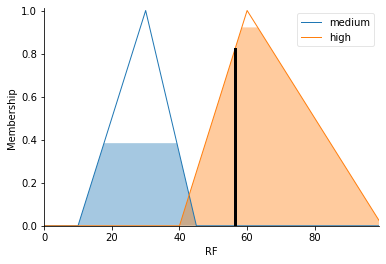
print("su riesgo finaciero es medio, tenga precaucion: ")

print()

print (tipping.output['RF'])

​

RF.view(sim**=**tipping)

su riesgo finaciero es Alto, Peligro: 56.464073711763675

Ejemplo2:

