



### Examen Final

Nombre: Fabian Armijos

Objetivo:

- Consolidar los conocimientos adquiridos en clase de los sistemas expertos.

Enunciado:

Se desea generar un sistema de recomendación de películas, por tal motivo se va a utilizar una base de datos orientada a grafos y un control de lógica difusa para clasificar el riesgo financiero, el mismo que será ingresado como atributo del cliente en el sistema recomendador, 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/1%C3%B3gica-difusa-ejercicios-propuestos-b99603ef1bc0>.
- 2) Generar números aleatorios para la edad y el porcentaje de manejo con el objetivo de generar al menos 100 personas y además incluir el listado de **películas** vistas y el valor del rating de cada película. Al menos 20 películas y un total de nodos de al menos 250 nodos.
- 3) Con estos datos aplicar el algoritmo de KNN y Similitud de Coseno para la recomendación de películas, seguir el siguiente tutorial: <https://guides.neo4j.com/sandbox/recommendations> o <https://github.com/MNoorFawi/recommendation-engine-with-neo4j> o <https://neo4j.com/developer/example-project/>.
- 4) Finalmente realizar alguna interfaz para poder acceder a la recomendación e ingreso de datos y resultados de los procesos.
- 5)





# Sistemas Expertos

## Tema: Basados casos y lógica difusa.

### Examen Final

neo4j@bolt://localhost:11003/neo4j - Neo4j Browser

File Edit View Window Help Developer

neo4j\$

Started streaming 962 records after 1 ms and completed after 45 ms.

neo4j\$ match (n) return n;

{"nombre": "Fast & Furious 8", "riesgo": 28.0}
{"nombre": "Cruella", "riesgo": 43.0}
{"nombre": "Minions: El origen de Gru", "riesgo": 49.0}
{"nombre": "The Forever Purge", "riesgo": 11.0}

neo4j\$ match (n) return n;

{"nombre": "Fast & Furious 8", "riesgo": 28.0}
{"nombre": "Cruella", "riesgo": 43.0}
{"nombre": "Minions: El origen de Gru", "riesgo": 49.0}
{"nombre": "The Forever Purge", "riesgo": 11.0}
{"nombre": "Space Jam: A New Legacy", "riesgo": 43.0}
{"nombre": "Luca", "riesgo": 82.0}
{"nombre": "Jungle Cruise", "riesgo": 58.0}
{"nombre": "El Escuadrón Suicida", "riesgo": 87.0}
{"nombre": "Halloween Kills", "riesgo": 37.0}
{"nombre": "Misión Imposible 7", "riesgo": 39.0}
{"nombre": "Matrix 4", "riesgo": 3.0}
{"nombre": "Proyecto X", "riesgo": 22.05738233397807}

Copy Ctrl + C  
Paste Ctrl + V

MAX COLUMN WIDTH:

26°C Soleado 21:08



# Sistemas Expertos

Tema: Basados casos y lógica difusa.

## Examen Final

Screenshot of a Jupyter Notebook titled "EXAMENFINAL\_SE" showing fuzzy membership functions and a GUI application.

The notebook displays two plots of fuzzy membership functions:

- Plot 1 (gure):** Shows membership functions for "gure" (green), "adulto" (orange), and "mayor" (blue) against "edad" (age). The x-axis ranges from 20 to 70, and the y-axis (Membership) ranges from 0.0 to 1.0.
- Plot 2 (manejo):** Shows membership functions for "bajo" (blue), "medio" (orange), and "alto" (green) against "manejo" (handling). The x-axis ranges from 0 to 100, and the y-axis (Membership) ranges from 0.0 to 1.0.

The notebook also shows code cells for `manejo.view()` and `riesgo.view()`, which trigger the GUI application.

The GUI application, titled "ExamenFinal", contains the following fields and buttons:

- Nombre:
- Edad:
- Porcentaje Manejo:
- Buttons: `calcDif`, `guardar`, `Recomendacion`
- Text area: `Hola q mas`
- Fuzzy output: `fuzzy: 46.80232924142274`

The background text in the notebook includes instructions for generating a PDF report and submitting scripts to a Git repository for evaluation.

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

Fecha de Entrega : 03/08/2021 - 13:55

Criterios de Evaluación

- Sistema lógico difuso: 30%



# Sistemas Expertos

Tema: Basados casos y lógica difusa.

## Examen Final

Autoguardado ☐ Examen-Enunc... Buscar (Alt+Q) Fabian Armijos Sarmiento

Archivo Inicio Insertar Dibujar Diseño Disposición Referencias Correspondencia Revisar Vista Ayuda Compartir Comentarios

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

Fecha de Entrega : 03/08/2021 - 13:55

Criterios de Evaluación

- Sistema lógico de inferencia
- Neo4J Knn: 30%
- Informe y resultados
- GUI, programación

Nota: Subir el sistema

ExamenFinal

Nombre:

Edad:

Porcentaje Manejo:

Hola q mas

fuzzy: 46.80232924142274

Página 3 de 3 281 palabras Español (España) Concentración 110 %

26°C Soleado ESP 21:10

### - Neo4J Knn: 30%

Graph Data Science Playground

File Edit View Window Help Developer

NEuler Home Run Single Algorithm Run Algorithm Recipe

Algorithm

K-Nearest Neighbors  
computes similarities between node pairs based on node properties

Projected Graph

Label

Relationship Type

Relationship Orientation

Algorithm Parameters

# of neighbors per nodes

Node Weight Property

Sample Rate

Delta Threshold

Random Joins

Help us improve NEuler

26°C Soleado ESP 21:13



# Sistemas Expertos

## Tema: Basados casos y lógica difusa.

### Examen Final

Graph Data Science Playground

File Edit View Window Help Developer

NEuler Home Run Single Algorithm Run Algorithm Recipe

All algorithms New algorithm +

SIMILARITY / K-NEAREST NEIGHBORS - STARTED AT 21:13:32

1. Configure 2. Results 3. Code

Table

Persona

From	Nodes
52.24050632911391	52.24050632911391 1 52.24050632911391 1 52.24050632911391 1 52.24050632911391 1 52.24050632911391 1 52.24050632911391 1
52.55441211260941	52.55441211260941 1 52.55441211260941 1 52.55441211260941 1 52.55441211260941 1 52.55441211260941 1 52.55441211260941 1
51.30603015075368	51.30603015075368 1 51.26766882516193 0.96 51.26766882516193 0.96 51.384215429928155 0.93 51.431038087344795 0.89 51.43387852261383 0.89
52.39195986882331	52.39195986882331 1 52.39195986882331 1 52.39195986882331 1 52.39195986882331 1 52.39195986882331 1 52.39195986882331 1

Help us improve NEuler

Graph Data Science Playground

File Edit View Window Help Developer

NEuler Home Run Single Algorithm Run Algorithm Recipe

Algorithm

K-Nearest Neighbors

computes similarities between node pairs based on node properties

Projected Graph

Label Pelicula

Relationship Type Any

Relationship Orientation Natural

Algorithm Parameters

# of neighbors per nodes 10

Node Weight Property riesgo

Sample Rate 0,5

Delta Threshold 0,001

Random Joins 10

Help us improve NEuler



# Sistemas Expertos

## Tema: Basados casos y lógica difusa.

### Examen Final

Graph Data Science Playground

File Edit View Window Help Developer

NEuler

Home Run Single Algorithm Run Algorithm Recipe

1. Configure 2. Results 3. Code

Table

Pelicula

From	Nodes
26	26 1 26 1 26 1 26 1 27 0.5 25 0.5 28 0.33 28 0.33 29 0.25 23 0.25
85	84 0.5 84 0.5 86 0.5 86 0.5 86 0.5 87 0.33 83 0.33 83 0.33 83 0.33 83 0.33
22	22 1 23 0.5 23 0.5 23 0.5 19 0.25 25 0.25 26 0.2 26 0.2 18 0.2 26 0.2
99	100 0.5 100 0.5 100 0.5 97 0.33 96 0.25 96 0.25 94 0.17 94 0.17 92 0.13 92 0.13
64	64 1 64 1 65 0.5 63 0.5 66 0.33 62 0.33 67 0.25 67 0.25 61 0.25 61 0.25
82	82 1 82 1 83 0.5 83 0.5 81 0.5 83 0.5 83 0.5 84 0.33 84 0.33 79 0.25
79	78 0.5 78 0.5 78 0.5 81 0.33 76 0.25 82 0.25 76 0.25 82 0.25 82 0.25 83 0.2
8	8 1 8 1 8 1 7 0.5 7 0.5 6 0.33 6 0.33 11 0.25 5 0.25

Help us improve NEuler

Graph Data Science Playground

File Edit View Window Help Developer

NEuler

Home Run Single Algorithm Run Algorithm Recipe

Generate Neo4j Browser Guide

You can generate a Neo4j Browser guide that contains the code to reproduce the algorithm run:

Send to Neo4j Browser

Run code fragments

Or you can reproduce the algorithm run by running the following code fragments:

Anonymous Graph Named Graph

An anonymous graph is created for the duration of the algorithm run. It is deleted before the algorithm returns its results.

```
:use neo4j;
```

Copy

```
:param limit => ( 42);

:param config => ({
  nodeProjection: 'Pelicula',
  relationshipProjection: {
    relType: {
      type: '+',
      orientation: 'NATURAL',
      properties: {}
    }
  },
  nodeWeightProperty: 'riesgo',
  topK: 10,
```

Copy

Help us improve NEuler



# Sistemas Expertos

## Tema: Basados casos y lógica difusa.



## Examen Final

Graph Data Science Playground

File Edit View Window Help Developer

NEuler Home Run Single Algorithm Run Algorithm Recipe

New algorithm +

ALGORITHM RUNS

**Similarity / K-Nearest Neighbors**  
Started at: 21:14:35

Node Projection Pelicula  
Relationship Projection \*, NATURAL

**Similarity / K-Nearest Neighbors**  
Started at: 21:14:09

Node Projection Pelicula  
Relationship Projection \*, NATURAL

**Similarity / K-Nearest Neighbors**  
Started at: 21:13:32

Node Projection Persona  
Relationship Projection \*, NATURAL

**Similarity / K-Nearest Neighbors**  
Started at: 21:12:29

Node Projection Pelicula  
Relationship Projection \*, NATURAL

Help us improve NEuler

neo4j@bolt://localhost:11003/neo4j - Neo4j Browser

File Edit View Window Help Developer

```
1 :param limit => ( 42);
2 :param config => ({
3   nodeProjection: 'Pelicula',
4   relationshipProjection: {
5     relType: {
6       type: '*',
7       orientation: 'NATURAL',
8       properties: {}
9     }
10  },
11  nodeWeightProperty: 'riesgo',
12  topK: 10,
```

\$ :play neuler/user-content-1ed46d77-c076-4547-b4ec-8676c42902a3.html

Setting parameters

We can setup the parameters used by the algorithm, by running the following statements:

```
@:param limit => ( 42);
:param config => ({
  nodeProjection: 'Pelicula',
  relationshipProjection: {
    relType: {
      type: '*',
      orientation: 'NATURAL',
      properties: {}
    }
  },
  nodeWeightProperty: 'riesgo',
  topK: 10,
```



### Examen Final

neo4j@bolt://localhost:11003/neo4j - Neo4j Browser

File Edit View Window Help Developer

neo4j\$

\$ :param limit => ( 42); :param config => ({ nodeProjection: 'Pelicula', relationshipProjection: { relType: { type: ...

neo4j\$ :param limit => ( 42)

neo4j\$ :param config => ({ nodeProjection: 'Pelicula', relationshipProjection: { relType: { type: '\*', or...

neo4j\$ :param communityNodeLimit => ( 10)

\$ :play neuler/user-content-1ed46d77-c076-4547-b4ec-8676c42902a3.html

### Setting parameters

We can setup the parameters used by the algorithm, by running the following statements:

```
@:param limit => ( 42);
:param config => ({
  nodeProjection: 'Pelicula',
  relationshipProjection: {
```

neo4j@bolt://localhost:11003/neo4j - Neo4j Browser

File Edit View Window Help Developer

```
1 CALL gds.beta.knn.stream($config)
2 YIELD node1, node2, similarity
3 WITH node1, collect({node: gds.util.asNode(node2), similarity: similarity}) AS to
4 RETURN gds.util.asNode(node1) AS from, to
5 LIMIT toInteger($limit)
```

An anonymous graph is created for the duration of the algorithm run. It is deleted before the algorithm returns its results. We would typically use this mode when playing around with different algorithms.

We can run the following code to run the algorithm and view the results:

```
@ CALL gds.beta.knn.stream($config)
YIELD node1, node2, similarity
WITH node1, collect({node: gds.util.asNode(node2), similarity: similarity}) AS to
RETURN gds.util.asNode(node1) AS from, to
LIMIT toInteger($limit)
```

3/4 < . . . >

neo4j\$ match (n) return n;

Graph

Table

Text

362

{

26°C Soleado

ESP 21:16





### Examen Final

neo4j@bolt://localhost:11003/neo4j - Neo4j Browser

File Edit View Window Help Developer

neo4j\$

```
neo4j$ CALL gds.beta.knn.stream($config) YIELD node1, node2, similarity WITH node1, collect({node: gds.util.asN...
```

from	to
<pre>{   "identity": 100,   "labels": [     "Pelicula"   ],   "properties": {     "nombre": "Avatar",     "riesgo": 26.0   } }</pre>	<pre>{   "node": {     "identity": 704,     "labels": [       "Pelicula"     ],     "properties": {       "nombre": "Dios de la Gerra",       "riesgo": 26.0     }   },   "similarity": 1.0 }</pre>

Started streaming 42 records after 95 ms and completed after 189 ms.

neo4j@bolt://localhost:11003/neo4j - Neo4j Browser

File Edit View Window Help Developer

neo4j\$

```
neo4j$ CALL gds.beta.knn.stream($config) YIELD node1, node2, similarity WITH node1, collect({node: gds.util.asN...
```

\*(154) Pelicula(154)

Displaying 154 nodes, 0 relationships.

\$ :param limit => ( 42); :param config => ({ nodeProjection: 'Pelicula', relationshipProjection: { relType: { type: ...

26°C Soleado 21:17



### Examen Final

```
neo4j@bolt://localhost:11003/neo4j - Neo4j Browser
File Edit View Window Help Developer

neo4j$
neo4j$ CALL gds.beta.knn.stream($config) YIELD node1, node2, similarity WITH node1, collect({node: gds.util.asN...

"from"                                     "to"
{"nombre":"Avatar","riesgo":26.0}          [{"node":{"nombre":"Dios de la Gerra","riesgo":26.0,"similarity":1.0},
{"node":{"nombre":"Mision Imposible","riesgo":26.0,"similarity":1.0},
{"node":{"nombre":"Rapidos y Furiosos","riesgo":26.0,"similarity":1.0},
{"node":{"nombre":"Dios de la Gerra","riesgo":26.0,"similarity":1.0},
{"node":{"nombre":"Fast & Furious 8","riesgo":25.0,"similarity":0.5},
{"node":{"nombre":"El Escuadrón Suicida","riesgo":27.0,"similarity":0.5},
{"node":{"nombre":"Fast & Furious 8","riesgo":28.0,"similarity":0.3333333333333333},
{"node":{"nombre":"Rapidos y Furiosos","riesgo":28.0,"similarity":0.3333333333333333},
{"node":{"nombre":"Space Jam: A New Legacy","riesgo":29.0,"similarity":0.25},
{"node":{"nombre":"Space Jam: A New Legacy","riesgo":23.0,"similarity":0.25}}]

{"nombre":"The Walking Dead","riesgo":85.0} [{"node":{"nombre":"Godzilla","riesgo":84.0,"similarity":0.5}, {"node":{"nombre":"The Forever Purge","riesgo":84.0,"similarity":0.5}, {"node":{"nombre":"The Forever Purge","riesgo":86.0,"similarity":0.5}, {"node":{"nombre":"Avatar","riesgo":86.0,"similarity":0.5}, {"node":{"nombre":"Matrix 4","riesgo":86.0,"similarity":0.5}, {"node":{"nombre":"Space Jam: A New Legacy","riesgo":83.0,"similarity":0.3333333333333333}, {"node":{"nombre":"Minions: El origen de Gru","riesgo":83.0,"similarity":0.3333333333333333}, {"node":{"nombre":"El Escuadrón Suicida","riesgo":83.0,"similarity":0.3333333333333333}, {"node":{"nombre":"Cruella","riesgo":83.0,"similarity":0.3333333333333333}, {"node":{"nombre":"El...

MAX COLUMN WIDTH: 100%
```

- Informe y resultados: 20%
- GUI, programación y pruebas: 20%

**Nota:** Subir el sistema en un cuaderno de Python + scripts + PDF.