

Informe Proyecto 2 - Fase 2
Sistema de Recomendación
“Desarrollo de un Sistema de Recomendaciones para
Universidades”
Grupo No. 7

Editado por:

Abner Gabriel Mejicanos Hernandez - 231134

Alejandro Rivera Rodriguez - 23674

Karen Daniela Pineda Ventura - 231132


María José Yee Vidal - 231193

Sistema de recomendación de universidades


Repositorio en GitHub:

https://github.com/agabriel01314/proyecto_final_datos

Documento de la fase 1:

 Proyecto 2 - Fase 1

Base de datos inicial:

 base de datos para datos jsjsjsj

Videos de usuarios de prueba:

[Enlace a videos](#)

GUÍA PARA UTILIZAR EL PROGRAMA DE RECOMENDACIÓN PARA EL USUARIO

1. INTRODUCCIÓN

Descripción del Proyecto

En varias ocasiones muchos jóvenes enfrentan una gran complicación al tratar de decidir qué universidad elegir para continuar con sus estudios superiores, esta decisión es crucial y puede impactar significativamente su crecimiento tanto personal como profesional.

Elegir la universidad adecuada no es una tarea sencilla, ya que se deben de considerar una variedad de factores importantes que incluyen: costos, intereses, la estructura, horarios, entre otros.

2. INSTALACIÓN

Requisitos Previos

Java Development Kit (JDK): Versión 8 o superior.

Neo4j: Versión 4.2.5.

Sistema Operativo: Windows, macOS o Linux.

Instalación de Neo4j

1. Descargar e instalar Neo4j:
 - Visita Neo4j Download Center <https://neo4j.com/download-center/>
 - Descarga la versión según su sistema operativo.
 - Sigue las instrucciones de instalación proporcionadas en la página.
2. Iniciar el servidor Neo4j:
 - Abre la consola/terminal.
 - Ejecuta el comando para iniciar el servidor.

3. CONFIGURACIÓN

Configuración del Archivo 'application.properties'

neo4j.uri: bolt:// localhost: 7687.

neo4j.username: neo4j.

neo4j.password: 123456789

Descargar librerías necesarias:

Pip install scikit-learn

Pip install pandas

Pip install neo4j

4. USO DEL SISTEMA

Inicialización del programa

1. Descargar base de datos que se adjunta en el presente documento.
2. Abrir la aplicación de programación de preferencia (Se recomienda al usuario el uso de Visual Studio).
3. Conectar la base de datos con Visual studio [foto]
4. Al ser ejecutado se muestra el siguiente menú en la terminal:

```
Menu de Recomendaciones
1. Recomendaciones por costo
2. Recomendaciones por ranking
3. Recomendaciones por áreas
4. Salir
Selecciona una opción: █
```

5. Según la elección de usuario, pueden aparecer los siguientes submenús:
 - Al seleccionar la primera opción, devuelve la opción para ingresar “el costo máximo” que tiene establecido la mensualidad universidad.

```
Menu de Recomendaciones
1. Recomendaciones por costo
2. Recomendaciones por ranking
3. Recomendaciones por áreas
4. Salir
Selecciona una opción: 1
Ingresa el costo máximo: 3000
Shape de uni_filtrados_numeric_array: (6, 2)
Universidad: Universidad de San Carlos de Guatemala
Costo: 300
Ranking: 1
Datos: [300 1]

Universidad: Universidad Mesoamericana
Costo: 3000
Ranking: 8
Datos: [3000 8]

Universidad: Universidad Regional de Guatemala
Costo: 2500
Ranking: 11
Datos: [2500 11]

Universidad: Universidad Rural de Guatemala
Costo: 2000
Ranking: 12
Datos: [2000 12]
```

- Al seleccionar la segunda opción, devuelve la opción para ingresar “el ranking” de la universidad.

```
Menu de Recomendaciones
1. Recomendaciones por costo
2. Recomendaciones por ranking
3. Recomendaciones por áreas
4. Salir
Selecciona una opción: 2
Ingresa el ranking: 3
Shape de uni_filtrados_numeric_array: (3, 2)
Universidad: Universidad Rafael Landívar
Costo: 4000
Ranking: 3
Datos: [4000 3]

Universidad: Universidad del Valle de Guatemala
Costo: 7000
Ranking: 2
Datos: [7000 2]

Universidad: Universidad de San Carlos de Guatemala
Costo: 300
Ranking: 1
Datos: [300 1]

Universidades recomendadas: ['Universidad Rafael Landívar' 'Universidad del Valle de Guatemala'
'Universidad de San Carlos de Guatemala']
```

- Al seleccionar la tercera opción, devuelve la opción para ingresar “el área” para buscar entre las carreras de las universidades.

```
Menu de Recomendaciones
1. Recomendaciones por costo
2. Recomendaciones por ranking
3. Recomendaciones por áreas
4. Salir
Selecciona una opción: 3
Ingresa el área: Medicina
Shape de uni_filtrados_numeric_array: (1, 2)
Universidad: Universidad de San Carlos de Guatemala
Costo: 300
Ranking: 1
Datos: [300 1]

Universidades recomendadas: ['Universidad de San Carlos de Guatemala']
```

5. REQUISITOS DEL SISTEMA

Hardware

Procesador: Mínimo 2 GHz dual-core.

Memoria RAM: Mínimo 4 GB, es recomendable 8 GB o más.

Almacenamiento: Mínimo 10 GB de espacio libre.

Software

Java Development Kit (JDK): Versión 8 o superior.

Neo4j: Versión 4.2.5.

Sistema Operativo: Windows, macOS o Linux.

Para realizar el sistema de recomendación de universidades se utilizó el algoritmo k-NN que se explica a continuación su funcionamiento

6. ALGORITMO k-NN (k-Nearest Neighbors)

El algoritmo k-NN (k-Nearest Neighbors) es un método que se utiliza para clasificar y para regresión. Todo su funcionamiento se basa en encontrar los k puntos de datos más cercanos con características similares a un punto de consulta, para luego realizar un tipo de predicción basada en las propiedades de estos vecinos.

Ventajas

1. Su simplicidad: el algoritmo k-NN es fácil de entender y de implementar.
2. Sin Suposiciones: No hace suposiciones sobre la distribución de los datos, a diferencia de otros algoritmos como la regresión lineal, sino que se basa de un .

Desventajas

1. Eficiencia: El algoritmo puede llegar a ser ineficiente para implementarse en grandes conjuntos de datos debido a la necesidad de calcular las distancias entre todos los puntos de entrenamiento, haciendo que el programa se sobrecargue y se ralentice el proceso.
2. Escalabilidad: El rendimiento puede llegar a degradarse en altas dimensiones debido al problema de los diversos fenómenos que surgen al analizar y organizar datos de espacios de múltiples dimensiones .
3. Sensibilidad al Ruido: el algoritmo k-NN puede ser muy sensible al ruido o modificaciones en los datos y también al valor de k seleccionado.

Funcionamiento del Algoritmo k-NN

1. Representación de Datos:

Todos los datos se representan como un tipo de puntos en un espacio n-dimensional, donde cada dimensión corresponde a una característica del dato principal o seleccionado.

2. Elección del Valor de k:

El valor de k se toma como un parámetro clave para el algoritmo, pues representa el número de vecinos más cercanos que se considerarán para realizar la predicción. El tamaño del valor k afectará directamente al modelo, pues un valor pequeño puede presentar muy pocos vecinos, mientras que al tener valores grandes puede ser muy general y perder detalles del dato principal.

3. Medición de la Distancia:

La distancia entre los puntos se mide utilizando diferentes métricas, como las métricas de la distancia euclidiana (para datos numéricos continuos) y la distancia de Manhattan (para datos categóricos o discretos). Estos utilizan una determinada fórmula matemática para calcularse:

Distancia Euclidiana

$$d(p, q) = \sqrt{\sum_{i=1}^n (p_i - q_i)^2}$$

Distancia de Manhattan

$$d(p, q) = \sum_{i=1}^n |p_i - q_i|$$

4. Identificación de los Vecinos más Cercanos:

Para identificar un nuevo punto de consulta, se deben calcular las distancias entre el primer punto seleccionado y todos los puntos en el conjunto de datos de entrenamiento. Todos los k puntos con las distancias más pequeñas son seleccionados como los vecinos más cercanos.

5. Predicción

- Predicción para la clasificación:

La etiqueta del nuevo punto se determina por la mayoría de votos o encuentros entre los k vecinos más cercanos. Significa que se le asigna la clase más frecuente entre los vecinos.

- Predicción para la regresión:

La predicción del valor para que sea el nuevo punto se hace calculando la media de los valores de los k vecinos más cercanos.

7. REFERENCIAS

GraphEverywhere, E. (2021). Grafos | Qué son, tipos, orden y herramientas de visualización. GraphEverywhere.

<https://www.grapheverywhere.com/grafos-que-son-tipos-orden-y-herramientas-de-visualizacion/>

Díaz, B., & Díaz, B. (2022). #11 Sistemas de recomendación y modelos basados en grafos. ImpulsateK - Artificial intelligence, tools, insights and wisdom gleaned from the knowledge of others.

<https://impulsatek.com/11-sistemas-de-recomendacion-y-modelos-de-aprendizaje-basados-en-grafos/>

Peña, A. (2021). Base de datos orientada a grafos NEO4J. Formadores IT.

<https://formadoresit.es/base-de-datos-orientada-a-grafos-neo4j/#:~:text=Neo4j%20es%20una%20base%20de,el%20almacenamiento%20de%20los%20datos>.

M, G. P. (2023). Explicación de algoritmos y estructuras de datos de grafos con Ejemplos en Java y C++. freeCodeCamp.org.

<https://www.freecodecamp.org/espanol/news/explicacion-de-algoritmos-y-estructuras-de-datos-de-grafos-con-ejemplos-en-java-y-c/>

Martínez, M. (2022). Sistemas de recomendaciones basadas en grafos

<https://docta.ucm.es/rest/api/core/bitstreams/c736326f-5259-4fed-9e66-e176f6508e13/content>

Pérez, L. (2023). ¿Cómo elegir universidad? | Blog UNITEC. Universidad UNITEC.

<https://blogs.unitec.mx/vida-universitaria/la-unitec/10-razones-para-elegir-universidad/>