Trabajo Final

Curso: Complejidad Algorítmica

Universidad Peruana de Ciencias Aplicadas

Sección: CC41
Docente: Abraham Sopla Maslucán

Integrantes:
Ruiz Torres, Erick Hérnan (u202118946)
Alcántara Cruz Rodrigo Alonso(u202216698)
Conza Hualpa, Alexia Evelyn (u202118038)

June 22, 2024; 17:41

Contents

1 Descripción del problema

El problema que veremos a cabo en este proyecto se enfoca en las recomendaciones de autos eléctricos, el objetivo es brindar a todos los usuarios sugerencias personalizadas de autos en base a características como precio, año, modelo, etc. La recomendación de estos vehículos es fundamental en la venta online ya que ayuda a los usuarios a obtener su auto adecuado en base a sus necesidades. Como menciona Pigna (2023):

> "Los automóviles son ampliamente utilizados como un medio de transporte personal, permitiendo a las personas llegar a sus destinos de manera cómoda y rápida. Esto es especialmente valioso en áreas donde el transporte público es limitado o poco eficiente."

Para estar en contexto con este problema, podremos usar un sistema en grafos. Estos son estructuras de datos que permiten las relaciones entre distintos elementos de forma eficiente, ahora con esta información, podemos representar los autos con las características mencionadas anteriormente como los nodos del grafo y las relaciones se pueden representar con aristas. Esto permitiría modelar las similitudes de los vehículos o crear una comparativa de estos para que el usuario pueda elegir acorde a su presupuesto y necesidad. Imaginemos el siguiente caso: El usuario busca un automóvil de marca tesla del año 2022, el sistema puede usar algoritmos de búsqueda en el grafo establecido para poder encontrar mediante un filtro todos los automóviles disponibles que cumplan las especificaciones del usuario.

2 Descripción del conjunto de datos (dataset)

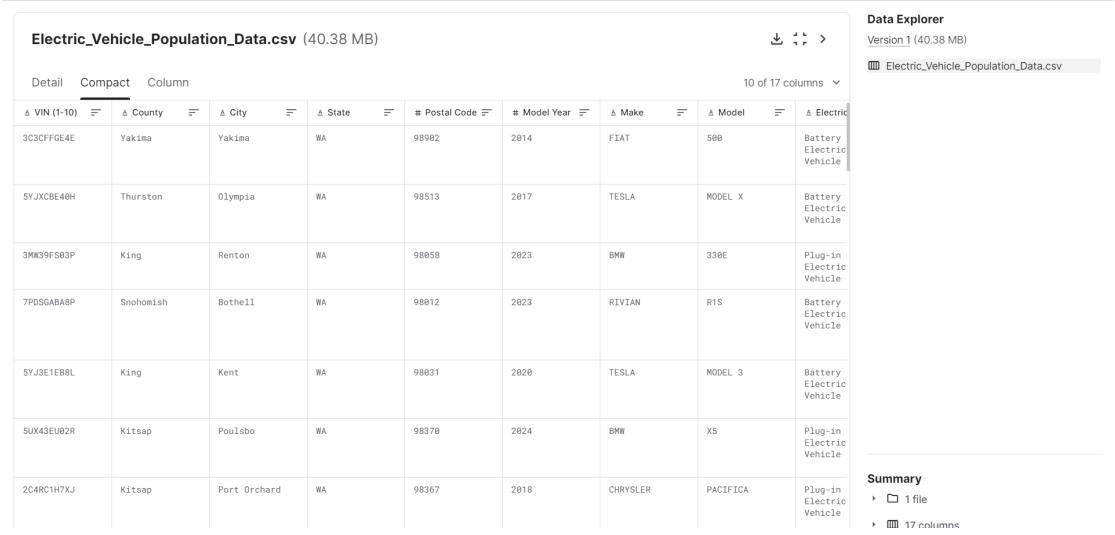
2.1 Características y origen de datos móviles de análisis

Vamos a abordar el problema mencionado anteriormente teniendo en cuenta ciertos criterios específicos, que incluyen el uso de un conjunto de datos creado para analizar y evaluar diferentes estrategias y soluciones. Por lo tanto, hemos desarrollado un conjunto de datos para recomendaciones de autos eléctricos, que contiene información detallada sobre autos eléctricos y sus características. En esta sección, se explicarán las características y el origen de los datos, además de presentar gráficos visuales que muestran la complejidad de las relaciones entre los autos eléctricos. El objetivo es proporcionar a los usuarios sugerencias personalizadas según sus preferencias y necesidades específicas, como la modelo, año del modelos, ubicación , etc. Este tipo de recomendación es crucial, ya que facilita a los clientes la búsqueda de vehículos eléctricos que se ajusten a sus necesidades individuales.

2.1.1 Origen de los datos

Los datos que se emplearán para el análisis de este proyecto principalmente han sido sacados de la plataforma en línea Kaggle. El nombre del dataset en línea es *Electric Vehicle Population Dataset* o, en su versión en español, *Conjunto de datos de población de vehículos eléctricos*. A su vez, se ha hecho generación de datos en algunas columnas de la base de datos final a utilizar en WattzFinder, el procedimiento detallado se explicará a continuación:

Al principio, se recopila información importante sobre las recomendaciones de autos eléctricos durante un período determinado.



Luego se traduce los nombres de las columnas para tener una base de datos

más organizada, debido a que nuestro proyecto Wattzfinder trabajará como un Sistema de recomendación basado en grafos se necesitará datos cuantitativos para poder utilizar diferentes métodos algorítmicos de grafos por datos cuantitativos, por ello se utilizará la biblioteca pandas para poder cargar la base de datos en un archivo csv en un DataFrame.

Uno de los principales problemas de nuestra Dataset sacada de Kaggle es que en la columna de Precios la mayoría valores son = 0, lo cual no tendría sentido para nuestro sistema de recomendación de carros electrónicos, por lo que en el siguiente código se extraerán y reemplazarán los precios que son = 0 con los datos originales de Kaggle que != 0, se utilizarán los datos previos y se distribuirán en las filas necesarias de manera aleatoria, a su vez se utilizará percentiles para mantener la distribución general de precios de la dataset original y completar la mayoría de datos vacíos en la columna Precios.

Luego se generarán 2 columnas adicionales por el mismo código "Descuentos" y "Precio Final", los descuentos se agregarán desde un 8% hasta un 40%, este descuento solo se aplicará a las filas con datos generados en Precios, es decir que los datos sacados en la dataset de Kaggle tendrán un descuento de 0%, esto se hará con el motivo que luego puedan ser identificables y observar si hay una correlación imprevista cuando hagamos el análisis final de nuestro proyecto y corregir si existe un error.

Aparte de utilizar la bibliotecas 'pandas' para los procesos relacionados con el archivo csv, se utilizará 'numpy' para el procesamiento de números en las 3 columnas modificadas en cuanto a la aleatoriedad y los percentiles.

En cuánto al código en general se utilizará las bibliotecas 'sys' y 'datetime' para traspasar el archivo con el script python como argumento y modificar su nombre con la fecha respectiva del procesamiento de datos.

El código python es el siguiente:

```python
import pandas as pd
import numpy as np
import sys
from datetime import datetime

def main(csv_file):
    df = pd.read_csv(csv_file)
    original_name = csv_file.split('.csv')[0]
    indices_precio_original_cero = df['Base MSRP'] == 0
    non_zero_msrp = df[df['Base MSRP'] != 0]['Base MSRP']
    percentiles = {
        'p10': non_zero_msrp.quantile(0.1),
```

```
        'p20': non_zero_msrp.quantile(0.2),
        'p30': non_zero_msrp.quantile(0.3),
        'p40': non_zero_msrp.quantile(0.4),
        'p50': non_zero_msrp.quantile(0.5),
        'p60': non_zero_msrp.quantile(0.6),
        'p70': non_zero_msrp.quantile(0.7),
        'p80': non_zero_msrp.quantile(0.8),
        'p90': non_zero_msrp.quantile(0.9)
    }

    def reemplazar_cero_con_percentiles(row):
        if row['Base MSRP'] == 0:
            percentil_elegido = np.random.choice(list(percentiles.keys()))
            return percentiles[percentil_elegido]
        return row['Base MSRP']

    df['Base MSRP'] = df.apply(reemplazar_cero_con_percentiles, axis=1)
    df['Descuentos'] = 0.0
    discount_percentage = np.random.uniform(8, 40,
    ↪  sum(indices_precio_original_cero)) / 100
    discount_values = (discount_percentage *
    ↪  df.loc[indices_precio_original_cero, 'Base MSRP']).round()
    df.loc[indices_precio_original_cero, 'Descuentos'] =
    ↪  discount_values.astype(int)


    df['Precio Final'] = (df['Base MSRP'] - df['Descuentos']).round()
    df['Precio Final'] = df['Precio Final'].astype(int)
    ahora = datetime.now()
    cadena_fecha_hora = ahora.strftime("%Y-%m-%d_%H%M%S")
    nuevo_nombre_archivo = '{}_modified_{}.csv'.format(original_name,
    ↪  cadena_fecha_hora)
    df.to_csv(nuevo_nombre_archivo, index=False)
    print(f"{nuevo_nombre_archivo} ha sido guardado.")


if __name__ == "__main__":
    if len(sys.argv) < 2:
        print("Uso: python modify-csv.py <archivoNombre.csv>")
        sys.exit(1)
```

```
49      main(sys.argv[1])
50
51
52
```

2.1.2 Características de los nodos

El conjunto de datos utilizado proporciona información sobre una variedad de autos eléctricos y está compuesto por las siguientes características o atributos para cada elemento:

1. **VIN:** Un número de identificación del vehículo (VIN) es el código de identificación de un automóvil específico.
2. **País:** País dónde se fabrica.
3. **Ciudad:** Ubicación de Empresa fabricante de Autos Eléctricos.
4. **Código postal:** El código PIN es el sistema de numeración de la oficina postal utilizado por el servicio postal.
5. **Año del modelo:** Año en el que se fabricó el automóvil.
6. **Nombre de la empresa.**
7. **Modelo:** El número de identificación del vehículo (VIN) es un código único de 17 dígitos específico para cada vehículo.
8. **Tipo de vehículo eléctrico:** Generalmente nos referimos a tres tipos principales de vehículos eléctricos: vehículos eléctricos híbridos (HEV), vehículos eléctricos híbridos enchufables (PHEV) y vehículos eléctricos de batería (BEV).
9. **Vehículo de combustible alternativo limpio:** Se pueden utilizar aceites vegetales, como palma, soja, girasol, maní y oliva, como combustibles alternativos para los motores diésel. Como combustible alternativo, el aceite vegetal es uno de los combustibles renovables.

10. **Autonomía eléctrica:** Los vehículos totalmente eléctricos normalmente pueden recorrer entre 110 y más de 300 millas con una sola carga. Los PHEV normalmente pueden recorrer entre 15 y 60 millas solo con la energía de la batería; la capacidad del tanque de combustible determina su autonomía general porque el motor arranca cuando la batería se agota.

11. **MSRP base:** Los fabricantes establecen un precio base para un automóvil o vehículo sin productos o características adicionales. El precio minorista sugerido por el fabricante (MSRP) es el precio base más funciones adicionales.

12. **Distrito Legislativo:** La Legislatura de cada Estado puede estar compuesta por el Gobernador y la Legislatura Estatal. En algunos de los Estados la Legislatura estará compuesta por dos Cámaras, a saber, la Asamblea Legislativa y el Consejo Legislativo, mientras que en el resto habrá haya una sola Cámara, a saber, la asamblea legislativa.

13. **ID de vehículo del DOL:** Un número de identificación único para cada vehículo está presente en el conjunto de datos de Transacciones. Las transacciones realizadas en el mismo vehículo tendrán la misma identificación de vehículo del DOL.

14. **Ubicación del vehículo:** En los casos en que el vehículo fue diseñado para motores eléctricos, generalmente se ubican en la parte delantera y/o trasera entre las ruedas. Hay semiejes cortos que conectan la salida de los motores a las ruedas.

15. **Servicio Eléctrico:** Una corporación, persona, agencia, autoridad u otra entidad o instrumento legal alineado con las instalaciones de distribución para el suministro de energía eléctrica para uso principalmente del público.

16. **Área censal de 2020:** Las áreas censales son divisiones geográficas definidas para la tabulación de datos. Pueden ser divisiones geográficas de área pequeña relativamente permanentes de un condado o entidad estadísticamente equivalente definida para la tabulación y presentación de datos del censo decenal y otros programas estadísticos seleccionados.

17. **Descuento:** Descuentos disponibles del vehículo.

18. **Precio Final:** Precios final del vehículo eléctrico.

2.2 Propósito, limitaciones y alcances de los datos

El propósito de recopilar y analizar este conjunto de datos es proporcionar recomendaciones personalizadas y relevantes de autos eléctricos a los usuarios a

través de una plataforma web, con el fin de mejorar la satisfacción del cliente y aumentar las recomendaciones o ventas de la empresa. En este contexto, podemos explorar la siguiente pregunta: ¿Cómo podemos utilizar las relaciones entre los productos para ofrecer recomendaciones personalizadas a los usuarios?

2.3 Identificación y representación de los nodos mediante grafos

En nuestro proyecto, planeamos crear un grafo que consiste en 166 801 nodos, los cuales representarán las propiedades de cada elemento. Para representar esta estructura gráfica, los nodos se conectarán entre sí en función de su similitud en la categoría principal (auto eléctrico), utilizando el algoritmo que se enseñó en clase. Todo este proceso se llevará a cabo utilizando el lenguaje de programación Python.

3 Propuesta

3.1 Objetivo General

El objetivo de nuestro proyecto *WatzzFinder* es desarrollar un sistema de recomendación avanzado basado en algoritmos de búsqueda en grafos. Este sistema representará y almacenará datos sobre vehículos eléctricos y sus interrelaciones de manera visual, permitiendo a los usuarios seleccionar el vehículo más adecuado según sus preferencias personales. Las características clave de los nodos que se utilizarán incluyen:

- Año del modelo,
- Nombre de la empresa,
- Modelo,
- MSRP base,
- Servicio Eléctrico,
- Descuento,
- Precio Final.

3.2 Objetivos Específicos

- Implementar técnicas de recorrido en grafos, como BFS (Búsqueda por Amplitud) y DFS (Búsqueda por Profundidad), para explorar las conexiones entre diferentes nodos representando vehículos y características de usuario.
- Utilizar variantes de DFS, como DLS (Búsqueda por Profundidad Limitada) y IDS (Búsqueda por Profundidad Iterativa), para optimizar la exploración de grafos bajo diferentes condiciones de búsqueda.
- Implementar algoritmos de grafos optimizados para manejar y procesar eficientemente un el volumen de datos de usuarios y vehículos eléctricos.
- Modelar relaciones complejas y múltiples entre entidades, no solo de manera directa entre los usuarios ante los objetos de compras sino también capturar interacciones e influencias indirectas.
- Proveer una interfaz de usuario que sea fácil de usar y que ayude a visualizar y entender cómo se generan las recomendaciones.

4 Procesamiento y Metodologías

4.1 Justificación de la Metodología

Para este proyecto se estará utilizando técnicas de recorrido y búsqueda en grafos como la principal metodología debido a su eficiencia en manejar relaciones complejas. Se emplearán métodos específicos que incluyen:

- **Búsqueda por Profundidad Limitada (DLS)** y **Búsqueda por Profundidad Iterativa (IDS)**: Estos métodos se aplicarán para explorar eficazmente el espacio de nodos del grafo, permitiendo identificar conexiones profundas entre características de vehículos. Dado que se procesará una gran cantidad de nodos (más de 160,000), estos métodos heurísticos son útiles para controlar la profundidad de búsqueda en grafos potencialmente grandes, asegurando que el sistema sea tanto eficiente como capaz de encontrar soluciones óptimas bajo restricciones específicas.
- **Algoritmo de Dijkstra modificado (A-Star)**: Se utilizará para encontrar el camino más corto o de menor costo entre dos nodos, lo que permite calcular las recomendaciones más relevantes basadas en múltiples factores ponderados. Este enfoque es dependiente de las diferentes características del carro electrónico. A su vez, se ha observado que en algunas modificaciones del algoritmo se ha utilizado para sistemas de recomendación de películas (Lubos, Fritscher, & Kriz, 2013) de la siguiente manera:

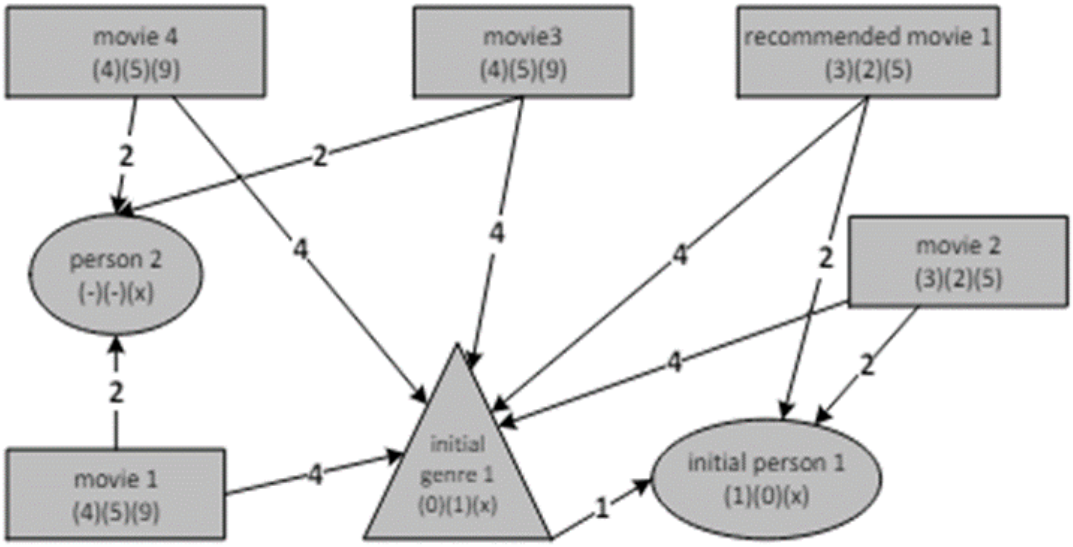


Figure 5. Visualization of Modified Dijkstra's algorithm.

Figure 1: Ejemplo de aplicación del algoritmo de Dijkstra modificado.

Aplicaciones de Algoritmos Avanzados

- **Algoritmo de Agrupamiento Espectral**: Basado en la descomposición espectral del Laplaciano del grafo (Javi, 2020), es excelente para capturar la estructura de datos de alta dimensión. Se utilizará para segmentar el mercado de usuarios, resultando en una segmentación final del mercado que depende de las dinámicas entre el usuario y el producto.

- **Algoritmo de Comunidad de Louvain**: Será efectivo para detectar comunidades dentro de grafos grandes y podría ser utilizado para identificar grupos de usuarios con preferencias similares y realizar recomendaciones grupales (Carazo & Amat, 2023). Este enfoque dependerá de la compra anterior de cada usuario.

La utilización de estos diferentes algoritmos en nuestro proyecto dependerá del objetivo específico, ya sea en recomendaciones basadas en múltiples factores específicos ponderados del mercado, segmentación de mercados o las previas compras de los usuarios.

4.2 Preparación y modelado de Datos

a. Adquirir los datos necesarios de la dataset de Kaggle *Electric Vehicle Population Dataset* con sus respectivas 16 características.

b. Escribir un programa en Python para generar datos faltantes de la columna MSRP base, aumentar 2 columnas adicionales de Descuento y Precio Final en el archivo ċsv de la dataset.

c. Construir el grafo donde cada nodo representa un vehículo eléctrico y las aristas reflejan las relaciones basadas en las 18 características / variables finales. Los usuarios también se modelarán como nodos.

4.3 Algoritmos de Grafos de búsqueda

- Implementación de UCS y A-Star para calcular las rutas más eficientes en costos y características, facilitando recomendaciones personalizadas.

- Aplicación de DLS y IDS para una exploración efectiva del grafo, identificando posibles recomendaciones basadas en la profundidad de búsqueda iterativa y limitada.

4.4 Implementación de código en Python

Se hará el uso de las siguientes librerías:

- **NetworkX:** Se utilizará NetworkX, para la creación, manipulación y estudio de la estructura, dinámicas y funciones de grafos complejos. Esta herramienta es ideal para manejar nuestro grafo de 166 801 nodos. También será utilizado para calcular métricas de grafos, como la centralidad y la detección de comunidades.
- **Numpy:** Se utilizará Numpy para la generación de datos faltantes en las columnas que originalmente vienen en la dataset de Kaggle y para la generación de datos para las nuevas columnas.
- **Matplotlib:** Se utilizará para crear gráficos estadísticos informativos para poder realizar retroalimentaciones basadas en Métricas de Evaluación. Además de realizar los grafos necesarios que complementen NetworkX.
- **Pandas:** Se utilizará para la manipulación y gestión del conjunto de datos. Esta biblioteca facilitará la adición de nuevas columnas como 'Descuentos' y 'Precio Final', permitiendo la manipulación precisa de datos tabulares. Además, será utilizada para aplicar transformaciones específicas como el llenado de datos faltantes y la implementación de operaciones condicionales basadas en los valores existentes. Se prevé que se continuará utilizando dicha biblioteca para la manipulación general de los datos, análisis y modelado posterior de la dataset.

5 Métricas de Evaluación

1. Porcentaje de Usuarios que realizan compras basadas en Recomendaciones Personalizadas.
2. Tiempo Promedio empleado en Búsqueda y recomendaciones calculando el tiempo medio que los usuario pasan desde que ingresan al sistema hasta que encuentren el vehículo finalmente comprado.

5.1 Big O evaluacion con tiempos

tomar capturas de pantalla y calcular con diferentes soluciones y cantidad de nodos.

5.2 Mejora continua y Evaluación de Feedback

- Ajuste de Algoritmos: Los algoritmos y heurísticas serán ajustados y afinados regularmente basándose en los resultados obtenidos de las métricas de evaluación. Sirve para mejorar continuamente la precisión y la eficiencia del sistema de recomendaciones.
- Feedback de Usuarios: Se establecerán canales de comunicación con los usuarios para recoger sus opiniones y sugerencias de manera sistemática.

6 Diseño del aplicativo

6.1 Diagrama de clase y diagrama de base de datos

6.2 Módulo para la adecuación de datos fuente a estructura de datos tipo grafo

6.3 Módulo para la representación de datos en el grafo

6.4 Interfaz tentativa (visual) para usuarios

7 Validación de resultados y pruebas

Validamos los resultados a través de..

8 Conclusiones

La recomendación de vehículos es crucial en las ventas online, ya que ayuda a los usuarios a encontrar el auto adecuado según sus necesidades individuales. El uso de un sistema en grafos permite modelar las relaciones entre diferentes autos de manera eficiente, representando las similitudes entre ellos y permitiendo una comparación fácil para que los usuarios elijan según su presupuesto y necesidad.

- La recomendación de vehículos es crucial en las ventas online, ya que ayuda a los usuarios a encontrar el auto adecuado según sus necesidades individuales.
- El uso de un sistema en grafos permite modelar las relaciones entre diferentes autos de manera eficiente, representando las similitudes entre ellos y permitiendo una comparación fácil para que los usuarios elijan según su presupuesto y necesidad.

- El sistema puede utilizar algoritmos de búsqueda en el grafo para encontrar todos los autos disponibles que cumplan con las especificaciones del usuario, como marca, año, y modelo.
- Los automóviles son esenciales como medio de transporte personal, especialmente en áreas con transporte público limitado o poco eficiente.

9 Referencias bibliográficas

- Pigna, A. (2023, 26 septiembre). *¿Para qué sirve el auto? Utilidad, función y ventajas*. Kavak. Available at: `https://www.kavak.com/mx/blog/para-que-sirve-el-auto-utilidad-funcion-y-ventajas`
- Carazo, F., & Amat, J. (2023, Abril). *Detección de comunidades en grafos y redes con python*. Available at: `https://cienciadedatos.net/documentos/pygml02-detecion-comunidades-grafos-redes-python`
- Javi, G. (2020, Agosto 28). *Spectral Clustering*. Available at: `https://javi897.github.io/Spectral_clustering/`
- Lubos, D., Fritscher, E., & Kriz, J. (2013). *Movie Recommendation Based on Graph Traversal Algorithms*. Available at: `http://www2.fiit.stuba.sk/~bielik/publ/abstracts/2013/televido-dexa2013.pdf`