Pontificia Universidad JAVERIANA Colombia

Tópicos Avanzados en Analítica
Proyecto II

Adriana Zuica Restrepo
David Oñate Acosta
Natalia Puello Acosta
Camilo Soto Zambrano
Diego Merlano Porto

1. Contexto

En el mundo actual del comercio electrónico, comprender las relaciones entre productos es fundamental para el éxito de las plataformas como Amazon. Los consumidores, al navegar por una tienda virtual, desarrollan un sentido de conexión entre productos basado en su apariencia, características y experiencias compartidas en las reseñas. La capacidad de entender estas relaciones, tanto visuales como basadas en opiniones, resulta crucial para guiar las decisiones de compra de los clientes y mejorar su experiencia general.

Amazon aprovecha el poder de los modelos de grafos y los sistemas de recomendación para revelar las conexiones entre productos y satisfacer las necesidades de sus clientes. Los modelos de grafos permiten representar visualmente las relaciones entre productos y usuarios, facilitando la comprensión de cómo se interrelacionan y complementan los elementos del catálogo de Amazon. Estos modelos sirven como base para construir sistemas de recomendación inteligentes.

Los datos de las plataformas de comercio electrónico, como la base que se utilizara a continuación, tienen una estructura de grafo natural. Los productos, usuarios, transacciones y sus relaciones se pueden representar como nodos y enlaces en un grafo. Esta representación facilita la captura de patrones y relaciones complejas, permitiendo predecir compras futuras de usuarios o identificar nuevas conexiones relevantes para el negocio, como la recomendación de productos no obvios.

2. Adquisición de Datos

Con este ejercicio se pretende usar grafos, desarrollar un análisis de varios productos de alimentos gourmet disponibles en Amazon, identificando la percepción de los usuarios frente a estos a través del tiempo y estableciendo las mejores recomendaciones posibles según el contenido de las reseñas y la utilidad de otros consumidores sobre las calificaciones. Para esto se tomará como referencia una base de datos recopilada de Amazon por J. McAuley y J. Leskovec en 2013 y disponible en el siguiente enlace:

https://snap.stanford.edu/data/web-FineFoods.html

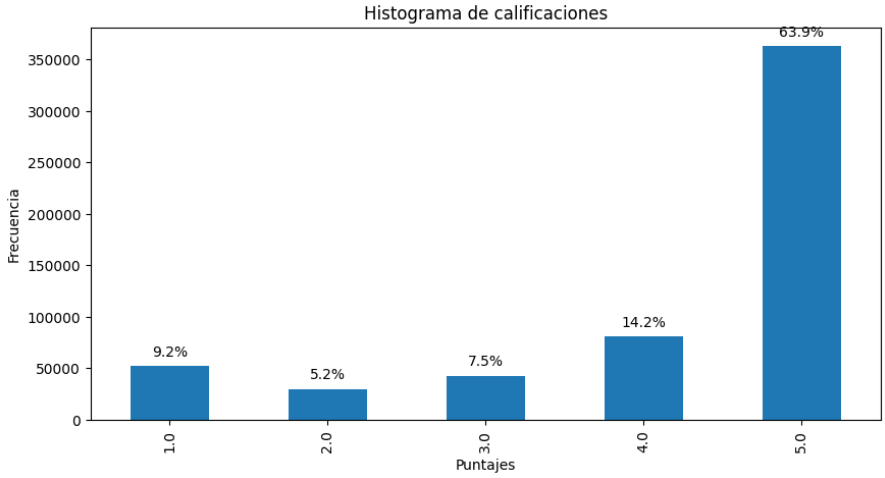
Inicialmente, se hará una revisión de la base de datos obtenida para lograr un entendimiento sobre las variables disponibles para el análisis, identificar si se requiere hacer transformaciones adicionales y/o tratamiento de valores nulos.

Tabla 1. Diccionario de datos			
Variable	**Tipo**	**Rango**	**Descripción de las variables**
ProductID	Texto		Identificador ASIN (particular de Amazon), alfanumérico limitado a 10 caracteres
UserID	Texto		Identificador del usuario
Profilename	Texto		Nombre del perfil del usuario
HelpfulnessNumerator	Numérica	0 – 866	Número de usuarios que encontraron útil la reseña consultada
HelpfulnessDenominator	Numérica	0 – 923	Número total de usuarios que calificaron la reseña
Score	Numérica	1 – 5	Calificación del producto en escala de 1-5
Time	Numérica	*	Serie de 14 años, tiempo en formato UNIX
Summary	Texto		Resumen de la reseña
Text	Texto		Texto completo de la reseña

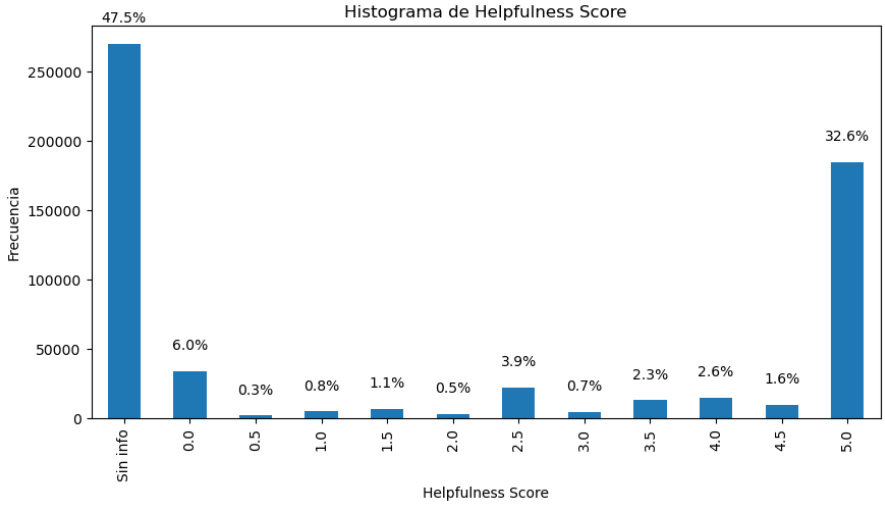
De esta primera revisión de los datos se identifican inicialmente dos transformaciones a realizar para poder hacer un entendimiento más profundo de las variables. En primer lugar, se observa que la utilidad de las reseñas viene separada en numerador y denominador, valores que de forma independiente no otorgan una medida estándar o fácilmente interpretable sobre la validez de la reseña; por lo cual se crea una nueva variable como la razón entre ambos datos que tomará valores entre 0 y 1 y permitirá una lectura más adecuada de este criterio. En segundo lugar, se observa que el tiempo viene en formato UNIX, el cual representa numéricamente los segundos transcurridos desde una fecha puntual; por ejemplo, la representación de las 23:59:59 del 31 de diciembre de 2023 tiene asignado el valor numérico 1704085199[1], por ende, para poder realizar interpretaciones adecuadas de acuerdo con la fecha de la reseña se requiere transformar esta columna en formato de fecha estándar.

Contamos con 74.258 productos y 256.059 usuarios únicos escribiendo reseñas en estos alimentos gourmet. Al analizar estas calificaciones, encontramos que el 63% de estas son puntuaciones perfectas, seguidas por 4 con un 14%.

[1] Múltiples conversores de tiempo estándar a tiempo UNIX pueden encontrarse en distintas páginas web. En este caso se utilizó el disponible en https://www.cdmon.com/es/apps/conversor-timestamp



Sin embargo, al revisar la utilidad de estas calificaciones, encontramos que el 48% de estas no tienen puntuación de este tipo, seguido por el 5 con 32%.



Analizando nuestras reseñas, observamos un rango de fechas desde 1999 hasta 2012. El año con mayor cantidad de reseñas fue el 2012, representando el 35% del total, seguido de cerca por el año 2011, con un 29%. En cuanto a los meses y días de la semana, no encontramos picos significativos de reseñas, lo que sugiere una distribución uniforme entre estos.

2.1 Visualización del grafo de interacciones

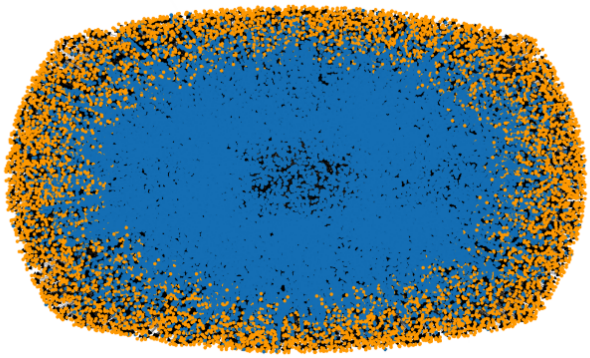
Es importante conocer los tipos de enlaces y nodos que contiene el grafo, para el caso de este se tiene lo siguiente:

```
Dataset: Amazon Food Reviews
----------------------------
Numero de grafos: 1
Cantidad de Nodos: 37948
Cantidad de features: 745
Cantidad de clases: N/A

Graph:
------
Se tienen links dirigidos: False
Grafo tiene nodos aislados: False
Grafo tiene self-loops: False
Cantidad de nodos aislados: 0
Cantidad de nodos con grado = 1: 30653
```

- No se cuenta con enlaces dirigidos
- No se cuenta con nodos aislados
- Hay 30.653 nodos que tienen un grado de 1, lo que significa que están conectados a solo otro nodo. Esto podría indicar que una gran cantidad de nodos representan usuarios que solo han interactuado con un solo producto o viceversa.

Se presenta una visualización del grafo de interacciones donde los nodos azules representan los usuarios y los nodos naranjas los productos. Las aristas indican las interacciones entre usuarios y productos, es decir todas las reseñas realizadas.



Sin el peso de las aristas (score)

Con peso de las aristas (score)

Tanto los nodos como las aristas están distribuidos de forma uniforme. Sin embargo, el grafo con los pesos de las aristas es ligeramente diferente y tiene áreas menos densas que otras demostrando la importancia de los mismos. Esta integración permite un análisis mas detallado de las interacciones proporcionando información relevante sobre las relaciones en el grafo.

2.2 Análisis de centralidad

La mayoría de los nodos tienen una centralidad de grado baja, esto se da por la naturaleza de la información donde hay nodos con muchas conexiones y muchos nodos con pocas conexiones.

Por último, se analizan comunidades, la dispersión de colores indica que las comunidades están bien mezcladas en el grafo, también hay interacciones cruzadas entre comunidades, lo que permite considerar productos de atractivo amplio y revisados por múltiples usuarios.

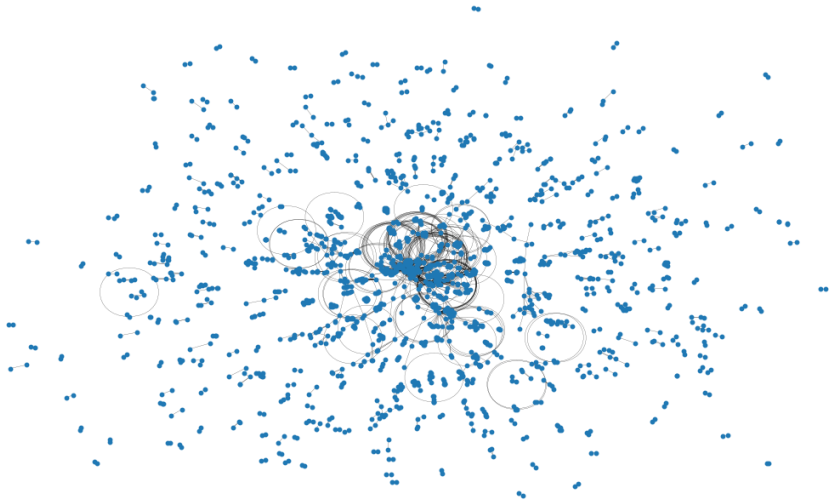
Modelos

3.1. Modelo Node2Vec – RecSys

Las primeras representaciones gráficas realizadas muestran la interacción entre los usuarios de Amazon y los productos reseñados a partir de la experiencia individual de cada cliente; lo cual brinda una oportunidad de análisis en la interacción entre ambas entidades; sin embargo, se decide iniciar con una representación algo más simplificada del problema, utilizando modelos de grafos enfocados en la recomendación de productos a partir de su frecuencia de ocurrencia conjunta.

Para este próposito, en primera instancia se aplicarán algunos criterios de filtrado que permitirán la obtención de una base de datos más adecuada para el ejercicio a realizar. En primera instancia, es improbable que un cliente compre productos con reseñas negativas, de la base original se conservarán solo aquellos registros con una puntuación asignada de mínimo 4 sobre una escala de 5 para luego definir la estructura del grafo, donde cada producto constituirá un nodo, y cada conexión con otros se establecerá si han sido reseñados en conjunto por más de 5 usuarios. Este criterio nos permite identificar un comportamiento de adquisición conjunta entre los productos disponibles en el catálogo de Amazon.

Ahora bien, la representación gráfica de los datos posterior al filtrado tiene un comportamiento de este estilo.



La utilidad de aplicar el Node2Vec es que a partir de la reducción en la dimensionalidad de los nodos y a partir de caminatas aleatorias, el modelo aprende secuencias de embeddings y aplica este aprendizaje para capturar posibles relaciones entre productos.

Teniendo en cuenta que este modelo no usa épocas, puesto que está basado en caminatas aleatorias y en la longitud de las mismas, se requiere definir una medida que permita capturar la eficiencia del modelo en la predicción; en este caso se usará la similitud de coseno, la cual permite identificar qué tanto se parecen dos productos a partir de sus embeddings a partir de una escala que va de -1 a 1, siendo este último la máxima similitud. Para generar un puntaje consolidado, se establece un umbral de 0.5 que servirá como frontera en la determinación de semejanza de dos productos y se tendrá al final una razón entre el total de correctos o similares sobre la magnitud de los registros evaluados.

Este primer acercamiento genera los siguientes valores de precisión:

	Validación	Prueba
Esc 1 (+)	0,8842	0,8731
Esc 2 (-)	0,8811	0,8696

Los siguientes modelos se redujeron a los primeros 40.000 primeros datos dado el procesamiento, de igual forma dado los resultados para el informe sólo se mostrará el modelo que tomo el peso de las aristas.

3.2. Modelo GCN - Graph Convolutional Network

Este modelo se seleccionó por su capacidad de capturar eficazmente las características de los grafos y las relaciones entre los nodos. Se busca predecir nuevas conexiones entre los nodos, es decir identificar posibles conexiones entre los usuarios y los productos.

```
Epoch: 010, Loss: 0.8513, Val: 0.7661, Test: 0.7637
Epoch: 020, Loss: 0.5370, Val: 0.8100, Test: 0.8091
Epoch: 030, Loss: 0.4280, Val: 0.8223, Test: 0.8213
Epoch: 040, Loss: 0.3594, Val: 0.8331, Test: 0.8316
Epoch: 050, Loss: 0.3442, Val: 0.8337, Test: 0.8280
Epoch: 060, Loss: 0.3246, Val: 0.8385, Test: 0.8345
Epoch: 070, Loss: 0.3196, Val: 0.8385, Test: 0.8345
Epoch: 080, Loss: 0.3066, Val: 0.8385, Test: 0.8345
Epoch: 090, Loss: 0.3077, Val: 0.8385, Test: 0.8345
Epoch: 100, Loss: 0.3108, Val: 0.8385, Test: 0.8345
```

El modelo muestra una mejora constante en la perdida de entrenamiento y la precisión en la etapa de validación y prueba alcanzando valores cercanos al 83%, lo cual nos permite asumir que realiza una buena predicción en los enlaces del grafo, es decir con las interacciones de reseñas de productos de Amazon. Sin embargo, consideramos que si tuviéramos más capacidad de cómputo podríamos aumentar el tamaño de la muestra mejorando los resultados del modelo.

	start_node	end_node
0	0	0
1	0	1
2	0	4
3	0	5
4	0	10

Se observa los posibles pares predichos por el modelo, en este caso el nodo 0 tiene múltiples conexiones, sin embargo, también tiene una auto conexión lo cual nos podría estar alertando de una anomalía en el grafo.

3.2.1 Variación hiperparámetros GCN

Con el fin de llegar al resultado mostrado en el numeral anterior, se hicieron variaciones al modelo entre las que se encuentran:

- Uso de LeakyReLu como función de activación
- Inclusión de weight decay
- Early stoping con variación en valor de patience
- Variación en la cantidad de capas convolucionales
- Capas de dropout para evitar sobre ajuste

El segundo mejor resultado para modelo GCN, se obtuvo incluyendo 3 capas convoluciones, con función de activación ReLu y capas de Dropout.

```
Epoch: 240, Loss: 0.3839, Val: 0.7467, Test: 0.6967
Epoch: 250, Loss: 0.3854, Val: 0.8067, Test: 0.7267
...
Epoch: 970, Loss: 0.3786, Val: 0.8233, Test: 0.7600
Epoch: 980, Loss: 0.3797, Val: 0.8233, Test: 0.7600
Epoch: 990, Loss: 0.3796, Val: 0.8233, Test: 0.7600
Epoch: 1000, Loss: 0.3735, Val: 0.8233, Test: 0.7600
```

3.3 Modelo GraphSAGE

Este modelo utiliza muestreo y agregación de características de los nodos vecinos para generar representaciones de nodos de manera eficiente y escalable. Mediante este modelo se espera mejorar la capacidad de predicción, lo que puede traducirse en recomendaciones más precisas y personalizadas.

Para llegar a los resultados, se realizaron varias optimizaciones y ajustes en el modelo GraphSAGE. A continuación, se presentan los detalles de la configuración final del modelo y las variaciones en los hiperparámetros que llevaron a mejorar su rendimiento.

Configuración del Modelo

El modelo GraphSAGE utilizado en el estudio se configura de la siguiente manera:

Parámetro	Valor
Capas de Convolución	3
Canales de Entrada	745
Canales Ocultos	128
Canales de Salida	64
Función de Activación	ReLU
Dropout	0.5
Optimizer	Adam
Tasa de Aprendizaje	0.005
Función de Pérdida	BCEWithLogitsLoss

Optimización de Hiperparámetros

Utilizando Optuna, se optimizaron los hiperparámetros del modelo. Los valores óptimos identificados fueron los siguientes:

Hiperparámetro	Valor Óptimo
Canales Ocultos	128
Canales de Salida	64
Tasa de Aprendizaje	0.005

Durante el proceso de optimización, se realizaron 5 pruebas para identificar las mejores combinaciones de hiperparámetros. A pesar de los ajustes realizados, el mejor resultado obtenido en términos de precisión fue del 56%.

3.4 Variational Graph Auto-Encoders (VGAE)

El autoencoder de grafos variacional (VGAE) es un marco innovador para el aprendizaje no supervisado en datos estructurados en grafos, basado en el concepto del autoencoder variacional (VAE). El uso de VGAE en nuestro problema de recomendaciones de productos en Amazon permite capturar las complejidades estructurales de las interacciones entre usuarios y productos, proporcionando representaciones latentes interpretables que mejoran la precisión del modelo. Este utiliza un encoder basado en redes convolucionales de grafos (GCN) para aprender representaciones latentes de los nodos en el grafo, que representan usuarios y productos. El modelo consta de tres capas de GCN que transforman las características de los nodos en un espacio latente, utilizando funciones de activación ReLU entre las capas

Los resultados del entrenamiento del modelo VGAE muestran una mejora inicial en la precisión y el rendimiento del modelo en las primeras épocas, con un ACC de validación que alcanza el 55% y un ACC de test de 50% hacia el final del entrenamiento. Sin embargo, hay signos de sobreajuste, ya que la Val Acc disminuye mientras la Test Acc se estabiliza en 50%.

```
Epoch: 010, Loss: 3.1072, Val Acc: 0.5100, Test Acc: 0.4950
Epoch: 020, Loss: 1.9284, Val Acc: 0.5500, Test Acc: 0.4750
Epoch: 030, Loss: 1.3173, Val Acc: 0.5067, Test Acc: 0.4750
Epoch: 040, Loss: 1.0277, Val Acc: 0.5067, Test Acc: 0.4750
Epoch: 050, Loss: 0.9430, Val Acc: 0.5500, Test Acc: 0.4750
Epoch: 060, Loss: 0.8759, Val Acc: 0.5000, Test Acc: 0.4750
Epoch: 070, Loss: 0.8426, Val Acc: 0.4867, Test Acc: 0.5000
Epoch: 080, Loss: 0.8295, Val Acc: 0.4733, Test Acc: 0.5000
Epoch: 090, Loss: 0.8245, Val Acc: 0.4467, Test Acc: 0.5000
Epoch: 100, Loss: 0.8118, Val Acc: 0.4267, Test Acc: 0.5000
```

3.5 Graph Attention Network

Graph Attention Networks (GAT) son una clase de redes neuronales que operan sobre grafos y utilizan mecanismos de atención para pesar la importancia de los nodos vecinos en cada operación de agregación. Esto permite que el modelo adapte dinámicamente la influencia de cada vecino basándose en las características del nodo.

El modelo GATModel se define con dos capas de GATv2Conv, que es una versión mejorada de la capa GAT original con una atención más efectiva y estable.

Los resultados del entrenamiento del modelo muestran varias tendencias y oportunidades que se pueden considerar para entender mejor el rendimiento del modelo y cómo se podría mejorar.

- Observamos que la pérdida del modelo decrece con el tiempo y la precisión aumenta, lo que indica que el modelo está aprendiendo de los datos a medida que avanza el entrenamiento.

- La precisión final es significativamente más alta que la inicial, sugiriendo que el modelo ha logrado capturar algo de la estructura y las relaciones en los datos.

- A medida que el entrenamiento progresa, el modelo parece estabilizarse, alcanzando una precisión más alta y consistente hacia los últimos epochs, lo que sugiere que puede haber alcanzado un punto de convergencia.

```
Epoch 1: Loss = 0.7390, Accuracy = 0.5254
Epoch 11: Loss = 0.6879, Accuracy = 0.5051
Epoch 21: Loss = 0.6936, Accuracy = 0.4909
Epoch 31: Loss = 0.6921, Accuracy = 0.4970
Epoch 41: Loss = 0.6904, Accuracy = 0.5254
Epoch 51: Loss = 0.6805, Accuracy = 0.5558
Epoch 61: Loss = 0.6764, Accuracy = 0.6004
Epoch 71: Loss = 0.6709, Accuracy = 0.6389
Epoch 81: Loss = 0.6710, Accuracy = 0.6775
Epoch 91: Loss = 0.6688, Accuracy = 0.6937
```

4. Solución propuesta

4.1 Modelo Node2Vec – RecSys:

El modelo busca una solución con un enfoque más simplificado a otros modelos mediante el enfoque a las conexiones entre productos. En este caso a partir de la reducción de la dimensionalidad y la revisión y análisis de embeddings establece similitudes entre pares de productos de acuerdo con la frecuencia de ocurrencia de compra conjunta. Los resultados del análisis muestran un Accuracy (definido a partir de la similitud de coseno media) por encima del 85%.

4.2 Modelo GCN - Graph Convolutional Network:

El modelo muestra una notable mejora en la pérdida y precisiones durante las primeras 60 épocas, después de lo cual se estabiliza, lo que sugiere que ha aprendido las características importantes de los datos, de igual forma la cercanía entre las precisiones de validación y prueba indica que el modelo tiene una buena capacidad de generalización.

El rendimiento del modelo GCN, con una precisión (accuracy) de hasta el 83.36% en las últimas épocas de entrenamiento, indica que es altamente eficaz para establecer conexiones entre los diferentes productos comprados por los usuarios de Amazon. Este resultado sugiere que el modelo puede ser utilizado como una base para un sistema de recomendación de productos eficiente en una plataforma de e-commerce.

Por último, tenemos la certeza de que al incrementar el conjunto de datos el modelo podría tener mayor rendimiento.

5. Bibliografía

- Kipf, T. N., & Welling, M. (2017a). Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907. https://arxiv.org/abs/1609.02907
- Kipf, T. N., & Welling, M. (2017b). Variational graph auto-encoders. arXiv preprint arXiv:1611.07308. https://arxiv.org/abs/1611.07308
- PyTorch. (2024). ReduceLROnPlateau. En PyTorch Documentation. https://pytorch.org/docs/stable/generated/torch.optim.lr_scheduler.ReduceLROnPlateau.html

- Variational Graph Auto-Encoders. https://arxiv.org/abs/1611.07308

6. **Reporte de tareas:**
   - **Adriana Zuica Restrepo:**
     1. Visualización de grafos de interacciones
     2. Análisis de centralidad de los grafos
     3. Modelos GCN
   - **Natalia Puello Acosta:**
     1. EDA
     2. Modelos GCN con variaciones de hiperparámetros, VGAE
   - **Camilo Soto Zambrano:**
     1. Descripción de datos
     2. Modelo Node2Vec – RecSys
   - **David Oñate Acosta:**
     1. Modelo GraphSage
     2. Ajuste de hiperparametros.
   - **Diego Merlano Porto:**
     1. Modelo GAT