Tópicos Avanzados en Analítica

Módulo 2: Grafos

Proyecto Final
Clasificación de Aeropuertos en Brasil

José David Narciso

Marysabel Mejia

Andres Felipe Daza

Rosana Torres

Monica Perez Morales

Bogotá

Facultad de Ingeniería
Maestría en Analítica para Inteligencia de Negocios

1. **Introducción:**

Los grafos, en el ámbito del análisis de datos y la inteligencia artificial, se han convertido en una herramienta esencial para representar y entender relaciones complejas y estructuras interconectadas. La versatilidad de esta estructura de datos permite modelar no solo relaciones entre elementos, sino también patrones de conectividad y dependencia en una amplia gama de aplicaciones. Una de las aplicaciones más notables de los grafos es el análisis de redes, que abarca desde las redes sociales hasta las infraestructuras de transporte. En particular, los grafos son especialmente útiles en la representación de redes de transporte, como las redes de rutas aéreas. Estas redes aéreas conectan aeropuertos de todo el mundo y desempeñan un papel crucial en la conectividad global y en la economía global. Comprender la conectividad y las relaciones entre los aeropuertos en una red de rutas aéreas es esencial para optimizar operaciones, tomar decisiones estratégicas y mejorar la eficiencia en la gestión de vuelos y pasajeros. El aprendizaje de representaciones vectoriales en grafos permite el análisis de nodos en función de sus relaciones y conectividad, lo que es fundamental para una amplia variedad de aplicaciones. En este informe, exploraremos la aplicación de varios modelos vectoriales para aprender representaciones de aeropuertos en base a su conectividad en la red de rutas aéreas de Brasil. Así mismo, implementaremos algunos modelos que nos permitirán predecir el nivel de tráfico de los aeropuertos, basado en el Este enfoque tiene un gran potencial para mejorar nuestra comprensión de la conectividad de aeropuertos en la región y optimizar la toma de decisiones en la gestión de rutas aéreas y la planificación de vuelos.

2. **Entendimiento del Problema:**

El transporte aéreo se ha convertido en uno de los medios de transporte más utilizado a nivel mundial debido a su facilidad de acceso, con viajes más rápidos y costos razonables. Su creciente demanda ha hecho posible lograr conectividad a casi todas las partes del mundo, con un número creciente de vuelos directos a las principales ciudades.

Para la Agencia Nacional de Aviación Civil de Brasil es importante conocer como se ha comportado la red de tráfico aérea y por consiguiente la actividad aeroportuaria a partir del número total de aterrizajes más despegues en un periodo determinado. En los últimos años han identificado un incremento en la demanda del servicio, sin embargo, no han garantizado los recursos suficientes en los aeropuertos con mayor actividad lo cual ha afectado los niveles de satisfacción de los pasajeros y los ha llevado a incurrir en costos de los cuales no se ha materializado un retorno efectivo.

En este sentido, la Agencia Nacional de Aviación Civil de Brasil a partir de la cantidad de vuelos ha

clasificado sus aeropuertos en cuatro categorías y le interesa asegurar los recursos técnicos, humanos y en infraestructura para cada uno de estos de acuerdo con la categoría definida, para de esta forma mejorar la experiencia de los pasajeros durante sus estancias en los aeropuertos.

3. Exploración de los Datos:

La data proviene del dataset Airports de PyTorch Geometric [1] y contiene información de la red de tráfico aéreo brasileña, la cual fue recopilada de la Agencia Nacional de Aviación Civil (ANAC) entre enero y diciembre de 2016. Se trata de un grafo dirigido con 131 nodos, que representan aeropuertos de Brasil, y 1074 aristas, que representan la existencia de rutas aéreas entre pares de aeropuertos. Cada nodo tiene un vector de características de dimensión 131. Las aristas son direccionales y pueden conectar un nodo consigo mismo. A cada uno de los aeropuertos se le asigna una etiqueta correspondiente a su nivel de actividad, medida en función del flujo de vuelos que maneja. En específico, la actividad del aeropuerto se mide por el número total de aterrizajes y despegues en el año correspondiente.

Cada aeropuerto tiene asignada una de las cuatro etiquetas posibles correspondientes a su actividad (0, 1, 2, 3). En particular, el conjunto de datos utiliza los cuartiles obtenidos a partir de la distribución empírica de la actividad para dividir el conjunto de datos en cuatro grupos, asignando una etiqueta diferente a cada grupo. Así, la etiqueta 1 se asigna al primer cuartil de los aeropuertos, en donde se encuentran el 25% menos activo, y así sucesivamente. Es importante destacar que todas las clases (etiquetas) tienen el mismo tamaño (número de aeropuertos). Además, las clases están más relacionadas con el papel desempeñado por el aeropuerto. En la Ilustración 1 se puede observar la distribución de las clases.

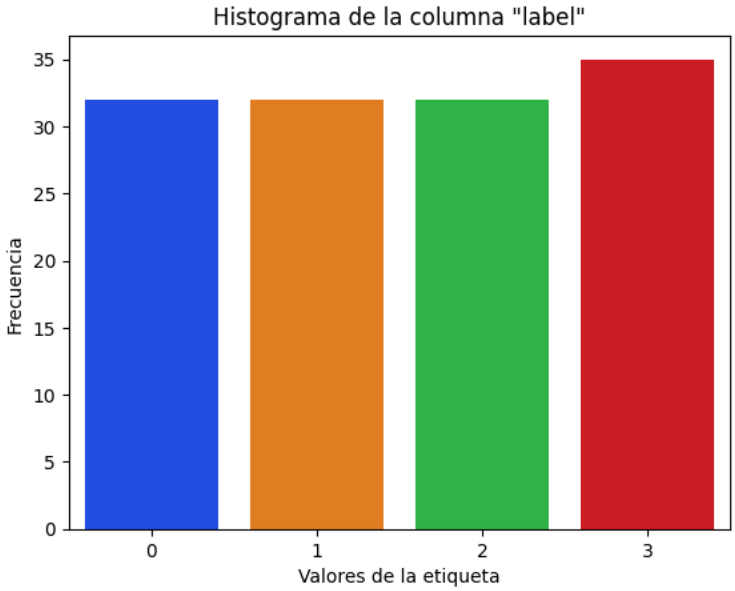

Ilustración 1: Distribución de las clases

En la Ilustración 2 se puede observar que el grafo está completamente conectado, no posee nodos aislados. Por otro lado, la Ilustración 3 graficó el grafo completo y filtrado (quitando aristas con bajo grado de nodo). Se aprecia que, si bien el grafo original está completamente conectado, al filtrar aristas surge una estructura con varios clusters fuertemente conectados internamente. Para obtener esta información gráfica de la data se implementó la librería NetworkX de Python.

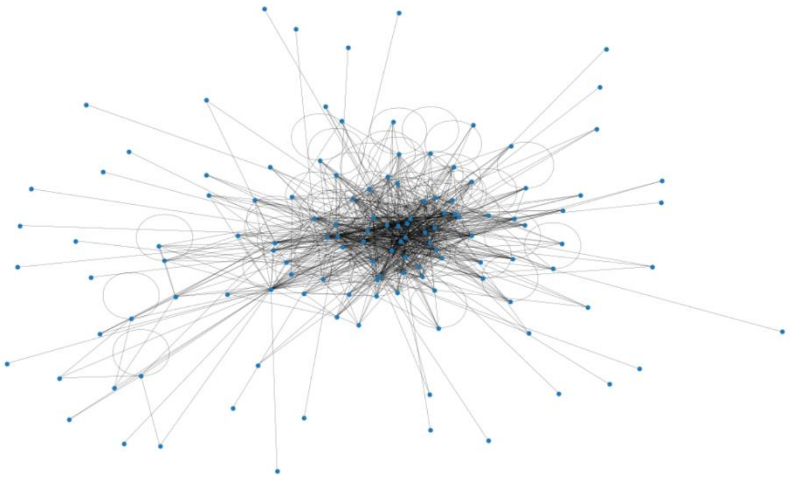

Ilustración 2: Interconectividad entre nodos

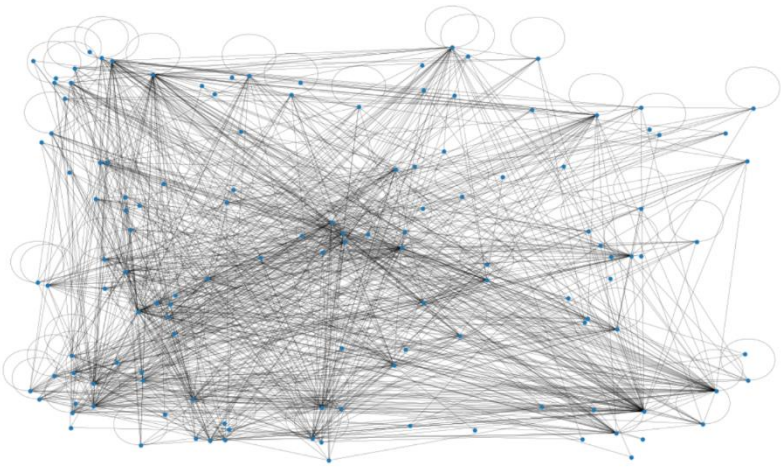

Ilustración 3: Grafico de los nodos con filtro de al menos 3 conexiones

A partir de la Ilustración 4, se logra analizar y visualizar la distribución de grados de los nodos. Se observa que el grado máximo es 81 y el grado mínimo es 1. La distribución tiene un sesgo hacia la izquiera, lo cual indica que la mayoría de nodos tienen un grado bajo.

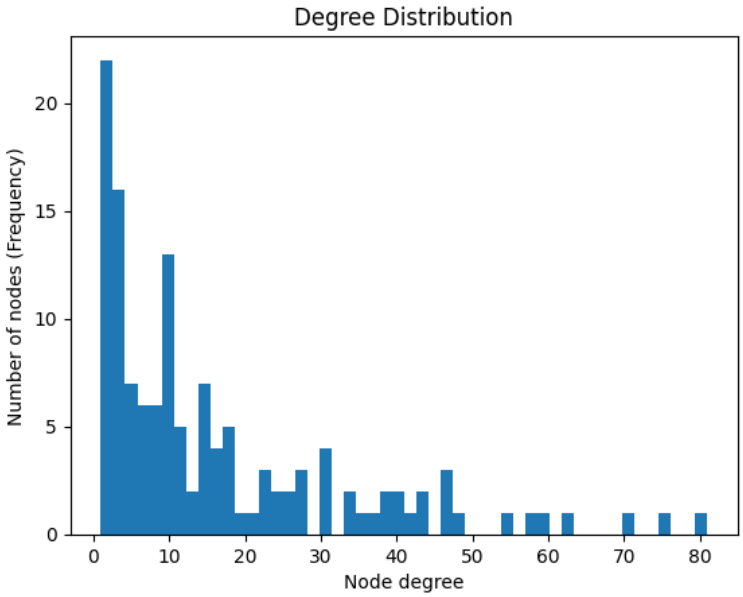

Ilustración 4: Histograma de distribución del grado de centralidad de la red

4. Modelos:

a. Modelo Graph Convolutional Networks (GCN):

Con el fin de realizar una segmentación de la data para la construcción del modelo, se dividió aleatoriamente el grafo en tres subconjuntos: entrenamiento (80 nodos), validación (14 nodos) y prueba (16 nodos). Esta

división se guardó en máscaras booleanas para indexar los nodos de cada subconjunto.

La GCN es una red neuronal diseñada específicamente para operar en estructuras de grafo. En este caso, se ha aplicado a una matriz de adyacencia que representa la conectividad entre nodos en una red de rutas aéreas de Brasil. El proceso de entrenamiento del modelo utiliza la función de pérdida de entropía cruzada (cross-entropy) y el optimizador Adam para ajustar los parámetros del modelo.

Hiperparámetros	**Valores que se usaron**
Tasa de aprendizaje (lr)	[0.001, 0.01, 0.1, 1]
Regularización (weight_decay)	[$1x10^{-4}$, $5x10^{-4}$,$1x10^{-3}$, $7x10^{-4}$]
Epocas (epochs)	[100, 200, 300]

Tabla 1: Hiperparámetros GCN

Para optimizar el rendimiento del modelo, se llevó a cabo una búsqueda exhaustiva de hiperparámetros, explorando tres aspectos clave: la tasa de aprendizaje (lr), el término de regularización (weight_decay) y el número de épocas (epochs). Se probaron diversas combinaciones de valores que se observan en la Tabla 1.

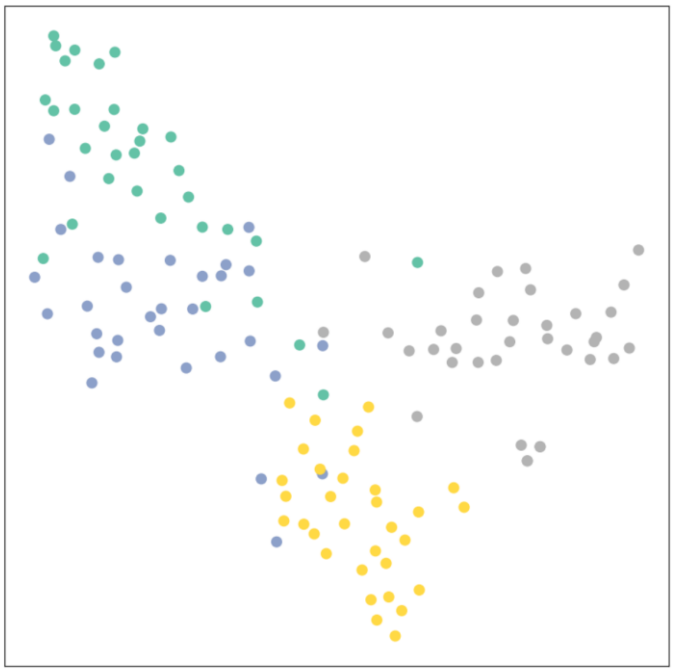

Ilustración 4: representaciones embebidas de los nodos aprendidas por la GCN

La mejor precisión en el conjunto de prueba fue 81.25%, con hiperparámetros: lr=1, weight_decay=0.0005, epochs=200. En la Ilustración 4 se aprecian claramente 4 clusters, que corresponden a las 4 clases de

aeropuertos. Esto demuestra que el modelo GCN fue capaz de aprender patrones de conectividad relevantes para la tarea de clasificación.

b. Modelo Node2Vec:

Por otro lado, usamos representaciones latentes para los nodos de la red de tráfico aéreo utilizando node2vec. Node2Vec es un algoritmo de aprendizaje de representación de nodos, el cual busca aprender representaciones vectoriales de nodos en un grafo de manera que los nodos similares en términos de estructura y conectividad en la red tengan representaciones vectoriales cercanas en un espacio de características de baja dimensión.

Estas representaciones vectoriales de nodos son útiles para diversas tareas, incluyendo la clasificación de nodos, el cual es el objetivo de nuestro problema de negocio. Node2Vec utiliza caminatas aleatorias para explorar el grafo y capturar la estructura local y global de la red. Dado que el modelo permite ajustar parámetros que controlan cómo se exploran las relaciones en el grafo, nos permitimos calibrar algunos de los hiperparámetros del modelo para obtener el mejor rendimiento posible acorde a nuestros datos.

Así entonces, se utiliza el modelo Node2Vec para aprender representaciones vectoriales de nodos en el grafo. Se configuran diversos parámetros como las dimensiones de los vectores resultantes, la longitud de los paseos aleatorios, el número de paseos aleatorios a realizar, así como parámetros que afectan la probabilidad de selección de nodos en el paseo aleatorio. La representación latente de cada nodo se convierte en la característica que luego se utiliza para entrenar un clasificador supervisado.

A continuación, los datos se dividen en conjuntos de entrenamiento y prueba utilizando train_test_split, y se construye un modelo de clasificación mediante el clasificador RandomForest. Se entrena el modelo y se evalúa su precisión (accuracy) en el conjunto de prueba. Luego, se repite el proceso utilizando un clasificador OneVsRest con regresión logística y máquina de soporte vectorial (SVM). Posteriormente, agregamos el grado de los nodos como variable predictora, ya que captura una noción muy básica de identidad estructural, y evaluamos los mismos modelos teniendo en cuenta esta modificación.

Finalmente, con el uso de la librería Optuna se procuró buscar y seleccionar los mejores hiperparámetros. Esta librería nos permitió explorar una amplia gama de hiperparámetros para cada uno de los modelos (Random Forest, Regresión Logísitca y SVM). Haciendo uso únicamente de los embeddings resultantes del modelo Node2Vec, obtuvimos que el mejor modelo es la máquina de soporte vectorial con C = 0.05. En este caso, obtuvimos un accuracy de 45%. Luego, implementando el grado del nodo como característica

predictora, adicional a la representación latente del nodo, obtuvimos un valor máximo de accuracy igual a 44.44% con el clasificador Random Forest ('max_depth' = 8, 'min_samples_split' = 15).

c. Modelo Graph Attention Networks (GAT):

Adicional, se ejecutó un modelo GAT como una red neuronal alternativa para representar nodos en grafos, a diferencia del modelo GCN, utiliza mecanismos de atención inspirados en Transformers y asigna pesos de atención diferentes a las conexiones con sus vecinos, permitiendo capturas relaciones específicas del grafo.

Hiperparámetros	**Valores que se usaron**
Tasa de aprendizaje (lr)	[0.001, 0.01, 0.1, 1]
Regularización (weight_decay)	[$1x10^{-4}$, $5x10^{-4}$,$1x10^{-3}$, $7x10^{-4}$]
Epocas (epochs)	[100, 200, 300]

Tabla 2: Hiperparámetros GCN

El modelo utiliza la función Exponential Linear Unit (ELU) y el optimizador Adam con los cuales se evalúan diferentes modelos con los hiperparámetros que están en la tabla 2. Como resultado se obtiene que el modelo con los hiperparámetros lr: 0.1, weight_decay: 0.0001, epochs: 200 permiten que el modelo alcance una precisión de 87.50%.

d. Modelo MLP:

El modelo MLP se plantea inicialmente una red neuronal con una capa de entrada del tamaño de los nodos del conjunto de datos, una capa de salida de tamaño 4 dado que son el número de clases que se tiene dentro de los datos, siendo estos los hiperparametros fijos, junto con el número de capas escondidas, el tamaño de estas y el número de épocas completan el modelo. Adicionalmente, para las capas ocultas se utiliza la activación RELU y en la capa de salida LOGSOFTMAX.

Hiperparámetros	**Valores que se usaron**
num_hidden_layers	[1,2,3,4,5,6]
dim_h	[32,64,128,256, dataset.num_features]
epochs	[100, 200, 300,500]

Tabla 3: Hiperparámetros MLP

Finalmente, al contar con un modelo que ya funciona, se ajustan los hiperparametros por de una búsqueda exhaustiva, donde se logra obtener una precisión del 68.75% en prueba, con 1 capa de escondida de 32 neuronas y 100 épocas.

5. Conclusiones:

Los resultados obtenidos a partir de los datos del tráfico aéreo en Brasil demuestran que la clasificación de sus aeropuertos es relativamente proporcional en sus cuatro categorías de acuerdo con el nivel de actividad aérea. Los resultados de las predicciones para los mejores modelos evidencian que la clasificación es adecuada, con una precisión del 87,5% en el modelo GAT. Esto último se configurará como el insumo necesario para que la Agencia Nacional de Aviación Civil (ANAC) de Brasil diseñe las estrategias necesarias que le permitan de manera anticipada asegurar los recursos necesarios para mejorar la experiencia de atención de sus pasajeros en los diferentes aeropuertos, pero en especial para los de mayor actividad. Además, se sugiere explorar la posibilidad de desarrollar modelos como struct2vec que de acuerdo con la literatura y las características de la red pueden sugerir resultados con predicciones más precisas.

6. Bibliografía:

[1] PyTorch Geometric dataset https://pytorch-geometric.readthedocs.io/en/latest/modules/datasets.html#airports