DETECCIÓN DE ANOMALÍAS TRANSACCIONALES APLICANDO TÉCNICAS DE MACHINE LEARNING CON GRAFOS

Presentado para obtener el título de

MAGISTER EN MATEMÁTICAS APLICADAS Y CIENCIAS DE LA COMPUTACIÓN

Juan Sebastián Cortés Sánchez

Dirección:
Juan Felipe Romero Ramirez

Universidad del Rosario
Escuela de Ingeniería, Ciencia y Tecnología
Maestría en Matemáticas Aplicadas y Ciencias de la Computación

Dedicatoria

Padre, esta tesis es el resultado de un largo y arduo camino, y quiero aprovechar este espacio para expresarte mi más sincero agradecimiento y dedicarte estas palabras. Tu apoyo incondicional, tu amor y tus enseñanzas han sido la base sólida que me ha impulsado a alcanzar este logro.

Gracias por estar a mi lado durante todas las etapas de mi educación. Tu constante apoyo emocional y económico han sido fundamentales para que pueda alcanzar este logro académico. Tus sacrificios y esfuerzos para brindarme las mejores oportunidades no han pasado desapercibidos, y valoro enormemente todo lo que has hecho por mí.

En esta dedicación, quiero reconocer el amor y la admiración que siento por ti. Hoy, con esta tesis en mis manos, quiero agradecerte de todo corazón por ser mi padre, mi mentor y mi amigo. Este logro también es tuyo, ya que sin tu apoyo y aliento constante, no habría llegado hasta aquí.

Agradecimientos

Agradezco de manera especial a mi tutor, Juan Felipe Romero, por su invaluable apoyo y orientación a lo largo de este proceso de investigación. Su experiencia, conocimiento y dedicación han sido fundamentales para el desarrollo de este trabajo.

También quiero expresar mi agradecimiento a mi compañero de trabajo, Julian Vallejo, por su invaluable apoyo y generosidad al brindarme su tiempo y conocimiento durante la etapa inicial de esta tesis.

Abstract

Este documento propone una metodología para la identificación de transacciones anómalas realizadas a través de un servicio de depósito electrónico de una entidad financiera con el objetivo de prevenir y detractar el lavado de activos y de financiación del terrorismo, esta metodología consiste en la implementación de múltiples técnicas de Machine Learning, especificamente de aprendizaje no supervisado.

Palabras Claves: *UIAF, Grafos, Detección de Anomalías, Isolation Forest, HBOS, ABOD, Análisis de Componentes Principales*

Índice

Lista de tablas

Lista de figuras

1. INTRODUCCIÓN

Actualmente, el mundo financiero enfrenta muchos problemas y amenazas relacionadas con el lavado de dinero y otras actividades financieras ilegales. Estas actividades representan una amenaza significativa tanto para las instituciones financieras como para la integridad del sistema económico conjunto. Es fundamental implementar medidas eficaces de prevención y detección para salvaguardar la integridad de las transacciones financieras y combatir el lavado de dinero.

La detección de anomalías transaccionales se ha convertido en una poderosa herramienta en la lucha contra el blanqueo de capitales. Esta técnica se basa en un análisis exhaustivo de grandes cantidades de datos financieros para identificar patrones y comportamientos inusuales que podrían indicar actividades sospechosas. La detección temprana de estas anomalías permite a las instituciones financieras y a las autoridades competentes tomar medidas preventivas y evitar la materialización de transacciones ilícitas.

Esta tesis tiene como propósito explorar los métodos y técnicas utilizadas en la detección de anomalías transaccionales y su aplicación en la prevención del lavado de dinero. Se analizará el espectro transaccional basado en teoría de grafos, así como la integración de diferentes herramientas de inteligencia artificial y aprendizaje automático en este campo.

El desarrollo de sistemas de detección de anomalías transaccionales efectivos y robustos se ha convertido en una prioridad para las instituciones financieras y las autoridades reguladoras. La implementación de estas herramientas no solo contribuye a la seguridad y confiabilidad del sistema financiero, sino que también es un paso crucial en la lucha contra el crimen financiero y el terrorismo.

En esta tesis se analizarán los datos de un servicio de depósito electrónico propio de una entidad financiera para un periodo de tiempo específico, y se propondrá una alternativa metodológica para fortalecer la prevención y detección del lavado de dinero. La investigación realizada en este campo tiene como objetivo contribuir al avance de las prácticas de seguridad financiera y brindar una mayor protección a las instituciones financieras y a la sociedad Colombiana en general.

2. OBJETIVOS

2.1. Objetivo general

Establecer alternativas metodológicas que permitan dar cumplimiento del Anti Money Laundering, generando confianza y lealtad de parte de los clientes, buscando eficiencias y garantizando la buena reputación de la organización.

2.2. Objetivos específicos

1. Identificar transacciones anómalas realizadas a través de un servicio de depósito electrónico propio de la organización.
2. Caracterizar a los individuos involucrados y/o participantes de transacciones anómalas para realizar los correspondientes reportes.
3. Disminuir la carga operativa en investigación de AML en alrededor de un 90 % dada una correcta aplicación de la metodología.

3. JUSTIFICACIÓN Y DESCRIPCIÓN DEL PROBLEMA

AML es el acrónimo de Anti-Money Laundering, conocido como "Prevención del Lavado de Dinero", tomando como base el término "Blanqueo de Capitales", el Grupo de Acción Financiera Internacional - GAFI, se define como la conversión o transferencia de propiedad, con pleno conocimiento de que deriva en un delito criminal, con el propostio de esconder o disfrazar su procedencia ilegal o ayudar a cualquier persona involucrada en la comisión del delito a evadir las consecuencias legales de su accionar. Este termino es principalmente utilizado en el sector financiero para referirse a aquellos controles que deben realizar las empresas para para evitar, identificar y reportar posibles conductas sospechosas relacionadas al lavado de dinero o blanqueo de capitales que puedan llevarse a cabo dentro de sus actividades.

Dentro del AML se desarrollan una serie de acciones, entre ellas, es indispensable establecer y determinar la identidad de los clientes, comprender la naturaleza de sus actividades económicas y evaluar riesgos de lavado de dinero dentro de ellas. De igual manera, entra a realizar el seguimiento y el análisis continuo de las transacciones para identificar patrones y operaciones sospechosas.

Los procedimientos AML abarcan leyes, regulaciones y acciones que tienen como objetivo evitar que los delincuentes disfracen fondos y los conviertan en activos obtenidos de manera legal.

Algunas de las acciones dentro de AML implican la verificación de antecedentes, la verificación de identidad, verificar que el dinero sea de fuentes lícitas antes de aceptar nuevos movimientos y transferencias, bloquear y prevenir operaciones que sigan patrones de lavado de dinero, reportar a los sistemas de prevención del Gobierno operaciones sospechosas, entre otras. Tomado de [1]

En Colombia existen una serie de normativas establecidas, entre ellas destacan las siguientes:

- La regulación al acceso, tratamiento y protección de información financiera.
- La prevención del lavado de activos por medio de decretos, leyes y sistemas.
- Creación de la Unidad de Información y Análisis Financiero (UIAF).

La Unidad de Información y Análisis Financiero (UIAF Colombia) un organismo de inteligencia financiera encargado de analizar los reportes de operaciones sospechosas (ROS) de lavado de activos.

La UIAF lidera el sistema de antilavado y contra la financiación del terrorismo de Colombia. Sus funciones giran alrededor de realizar inteligencia financiera, emitir normativas, fiscalizar su cumplimiento e imponer sanciones. Además, debe difundir información pública y promover la integración de actores públicos y privados para mejorar el análisis de amenazas.

Estas responsabilidades tienen como fin mitigar los riesgos de lavado y financiación del terrorismo con el fin de proteger a la economía ya la sociedad. Tomado de [2]

El SARLAFT, Sistema de Administración del Riesgo de Lavado de Activos y de la Financiación del Terrorismo, es un mecanismo desarrollado por el Banco de la República para dar cumplimiento a la Circular Básica Jurídica 029 de 2014 de la Superintendencia Financiera de Colombia. Ese sistema, aplica a todos los clientes o vinculados contractualmente con cualquier entidad, sea empresa privada u organismo público.

Este sistema está compuesto por dos partes: el componente de prevención del riesgo y el componente de control. El primero busca prevenir que las entidades vigiladas sean utilizadas para el lavado (es decir, de fachada de recursos provenientes de ilícitos) o para la financiación del terrorismo. El segundo implica detectar las operaciones que pretendan realizar estas actividades.

La Superintendencia Financiera establece que la gestión del riesgo se debe desarrollar siguiendo etapas:

- Identificación del riesgo: se trata de identificar aquellos riesgos relacionados o vinculados, en cualquiera de sus formas, con el lavado de activos o la financiación del terrorismo.

- Medición de la probabilidad y el impacto del riesgo: las empresas que dispongan de un SARLAFT, tienen que poder medir qué probabilidad existe de que el riesgo tenga lugar y, en caso de tener lugar, qué impacto generaría.
- Control del riesgo: además, se tienen que establecer una serie de medidas para el control de los riesgos identificados.
- Monitoreo del riesgo: en este punto, se considera el seguimiento de las medidas o de los indicadores específicos para medir el riesgo, manteniéndose alerta a actividades que puedan resultar sospechosas o inusuales.

Para ello se vale de instrumentos tales como políticas, procedimientos, documentación, estructura organizacional, órganos de control, infraestructura tecnológica, divulgación de la información y capacitación.

No obstante, cada SARLAFT es diferente, ya que no existe un modelo previamente establecido, sino simplemente unos cumplimientos necesarios. Por lo cual, cada empresa u organismo debe crear, desarrollar y perfeccionar el suyo, a modo de sistema para la gestión del riesgo, pero específico para el lavado de activos y la financiación del terrorismo. Tomado de [3]

El cumplimiento debe hacer parte de la cultura empresarial de cualquier entidad, apalancando en estrategias que disminuyan las operaciones de riesgo. Estas políticas de cumplimiento son indispensables para evitar sanciones, afectaciones sobre la calificación crediticia y proteger la reputación de la organización, sin embargo, van más allá, pretenden mejorar la eficiencia y añadir valor real a la compañía. Al potenciar las actividades de conocimiento sobre el cliente, la organización centra su atención en generar una mayor satisfacción, de esta manera la confianza de los clientes aumenta y se valora la intención de cumplimiento de las políticas.

Por lo anteriormente descrito, este documento pretende suplir las etapas de identificación del riesgo y medición de la probabilidad del riesgo descritas anteriormente, suministrar dicha información a las unidades expertas de la entidad bancaria para que cuenten con las herramientas necesarias para abarcar las etapas posteriores.

4. ESTADO DEL ARTE

La detección de anomalías transaccionales es una de las principales actividades en el análisis de datos financieros, en múltiples ocasiones se ha demostrado que el uso de técnicas de Machine Learning es efectivo para identificar anomalías en el comportamiento transaccional de individuos de toda naturaleza. A continuación se presentan una serie de documentos que fueron tomados como punto de referencia para el desarrollo del presente trabajo:

1. **"Isolation Forest"** por Liu, et al. (2008) [4]

   El Ïsolation Forest"(Bosque de Aislamiento) es un algoritmo de detección de anomalías que utiliza un enfoque basado en árboles de decisión para identificar instancias anómalas en conjuntos de datos.

   El objetivo principal de este algoritmo es separar las instancias anómalas del conjunto de datos general mediante la construcción de árboles de decisión aleatorios. A diferencia de otros algoritmos que se centran en encontrar grupos de instancias normales, el Isolation Forest se basa en la premisa de que las anomalías son más susceptibles de ser aisladas y separadas en el espacio de características.

   El proceso de construcción de los árboles de decisión se basa en la aleatoriedad. En cada paso, se selecciona una característica aleatoria y se elige un valor de corte aleatorio dentro del rango de esa característica. Luego, se divide el conjunto de datos en dos subconjuntos en función de la característica y el valor de corte seleccionados. Este proceso se repite recursivamente hasta que cada instancia se aísle en un nodo terminal del árbol.

   Una de las ventajas clave del Isolation Forest es su eficiencia computacional, ya que puede manejar grandes conjuntos de datos de manera eficiente. Además, no requiere asumir ninguna distribución subyacente en los datos y es robusto frente a valores atípicos y ruido.

2. **"Unsupervised Anomaly Detection in High-Dimensional Data: A Survey"** por Varun y Bollmann (2019) [5]

   En este documento se exploran múltiples métodos de detección de anomalías no supervisados en datos de alta dimensión. Tiene como objetivo presentar los diferentes enfoques existentes y analizar su eficacia y limitaciones. Este artículo inicia describiendo los retos asociados con la detección de anomalías en conjuntos de datos de alta dimensión y de la presencia de características poco relevantes. Luego se presentan y clasifican los métodos en tres categorías principales:

   - Métodos basados en densidad.
   - Métodos basados en subespacios.
   - Métodos basados en el análisis de componentes principales.

   En la categoría de métodos basados en subespacios se estudian enfoques como el análisis de componentes independientes (ICA) y el análisis de componentes principales robusto (RPCA), métodos que tratan de detectar anomalías mediante la identificación de subespacios donde los datos normales y los datos anómalos presentan comportamientos diferentes.

   Por otra parte se explora la implementación de PCA y algunas de sus variantes como Sparse PCA y Robust PCA que permiten identificar anomalías en datos de alta dimensión mediante la proyección de los datos en subespacios de menor dimensión.

   En resumen, este artículo presenta una visión general de los múltiples métodos de detección de anomalías existentes, los clasifica y evalúa su desempeño y limitaciones.

3. **"Anomaly Detection in E-commerce Transactions using Machine Learning Techniques"** por Chandola, et al. (2009) [6]

   Este artículo se centra en la detección de anomalías en transacciones de comercio electrónico utilizando técnicas de aprendizaje automático.El crecimiento exponencial del comercio electrónico ha llevado a un aumento en las transacciones en línea, lo que ha generado un aumento paralelo en las actividades fraudulentas y maliciosas. La detección de anomalías en transacciones de comercio

electrónico se ha convertido en un desafío crucial para las empresas y los proveedores de servicios en línea, ya que buscan salvaguardar la integridad y la seguridad de sus sistemas.

En este artículo, los autores exploran diferentes enfoques y algoritmos de aprendizaje automático utilizados para la detección de anomalías en transacciones de comercio electrónico. Comienzan describiendo los diferentes tipos de anomalías que pueden ocurrir en este contexto, como fraudes con tarjetas de crédito, actividades de phishing y transacciones sospechosas.

Luego, presentan una revisión exhaustiva de diversas técnicas de aprendizaje automático, como clasificación, agrupamiento y algoritmos de detección de anomalías. Estos enfoques se aplican a datos transaccionales para identificar patrones y comportamientos anómalos que podrían indicar actividades fraudulentas.

En resumen, el artículo proporciona una visión general de los enfoques y las técnicas utilizadas para la detección de anomalías en transacciones de comercio electrónico.

4. **"A Novel Hybrid Intrusion Detection System based on One-Class Support Vector Machines"** por Hodo, et al. (2017) [7]

   Este artículo presenta un nuevo enfoque para desarrollar un sistema de detección de intrusiones (IDS) utilizando una combinación de técnicas, centrándose específicamente en las máquinas de vectores de soporte de una clase (SVM).

   Los sistemas de detección de intrusiones son fundamentales en la seguridad de redes para identificar y prevenir el acceso no autorizado o actividades maliciosas. Los métodos tradicionales de IDS a menudo se basan en detección basada en firmas, lo que requiere una amplia base de datos de patrones de ataques conocidos. Sin embargo, estos métodos pueden tener dificultades para detectar ataques novedosos o desconocidos.

   El IDS híbrido propuesto en el artículo tiene como objetivo superar estas limitaciones al incorporar máquinas de vectores de soporte de una clase. Las SVM de una clase son algoritmos de aprendizaje automático utilizados para la detección de anomalías, especialmente cuando los datos etiquetados de instancias normales y anómalas son escasos o no están disponibles.

   El IDS híbrido combina las fortalezas de los enfoques de detección basados en firmas y basados en anomalías. Al incorporar las SVM de una clase, el sistema puede adaptarse a ataques previamente no vistos o variaciones en los patrones de tráfico de red.

   En resumen, el artículo presenta un nuevo enfoque para desarrollar un IDS utilizando SVM de una clase, con el objetivo de mejorar las capacidades de detección y adaptabilidad a ataques novedosos en la seguridad de redes.

Ahora, con el objetivo de complementar la metodología planteada se revisó un conjunto adicional de documentos, se presenta a continuación:

1. **"DeepLog: Anomaly Detection and Diagnosis from System Logs through Deep Learning"** por Du, et al. (2017) [8]: El enfoque de este trabajo se centra en detectar anomalías en registros de sistemas mediante el uso de redes neuronales profundas. Se propone una arquitectura de red neuronal llamada DeepLog, que utiliza una combinación de LSTM (Long Short-Term Memory) y autoencoders para identificar comportamientos anómalos en los registros de sistemas.

2. **"Robust Random Cut Forest Based Anomaly Detection on Streams"** por Guha, et al. (2016) [9]: Este artículo presenta un enfoque basado en Random Cut Forest (RCF) para detectar anomalías en datos de transacciones en tiempo real. RCF es un algoritmo basado en árboles que construye un bosque aleatorio y utiliza la profundidad de los árboles para detectar anomalías. Este enfoque se adapta bien a flujos de datos continuos y es robusto ante cambios en la distribución de los datos.

3. **"A Survey of Network Anomaly Detection Techniques"** por Patcha y Park (2007) [10]: Este trabajo ofrece una visión general de las técnicas de detección de anomalías en redes, incluyendo enfoques basados en machine learning.

4. **"Deep Learning for Anomaly Detection: A Survey"por Chalapathy y Chawla (2019)** [11]: Esta revisión proporciona una visión general de las técnicas de aprendizaje profundo utilizadas en la detección de anomalías, incluyendo redes neuronales convolucionales, redes recurrentes y autoencoders.

5. **"Anomaly Detection in Streaming Data: A Survey"** por Akoglu, et al. (2015) [12]: Esta revisión se centra en la detección de anomalías en flujos de datos en tiempo real, incluyendo técnicas basadas en aprendizaje automático, como SVM y redes neuronales recurrentes.

6. **"Autoencoders for Unsupervised Anomaly Detection in Network Traffic Data"** por Sultana, et al. (2018) [13]: En este estudio, se utiliza un enfoque basado en autoencoders para detectar anomalías en datos de tráfico de red sin etiquetas, logrando resultados prometedores en la detección de ataques.

7. **"Network Anomaly Detection using Recurrent Neural Networks" por Xu, et al. (2018)** [14]: En este estudio, se emplea una red neuronal recurrente para detectar anomalías en datos de tráfico de red, demostrando una mejora significativa en comparación con métodos tradicionales.

Como se puede observar la literatura complementaria trabaja con algoritmos de aprendizaje profundo y reforzado. Es evidente que el campo de investigación crece constantemente, razón por la cual existen muchas más técnicas. Ahora bien, es clave mencionar que la elección de la técnica o algoritmo debe estar basada en el contexto específico de los datos transaccionales a analizar.

5. MARCO TEÓRICO

5.1. Grafos

5.1.1. Introducción

Son estructuras matemáticas que se usan para describir y analizar individuos con relaciones y/o interacciones. Pueden ser utilizados como una herramienta de representación, por ejemplo, la información/conocimiento está organizada y relacionada, el software puede ser representado como un grafo, las redes de similitud que conectan puntos similares y estructuras relacionales como moléculas y formas 3D. También pueden ser utilizados como redes (Conocidos como Grafos Naturales), por ejemplo, las redes sociales, las comunicaciones y transacciones: Dispositivos electrónicos, llamadas telefónicas, transacciones financieras, la Biomedicina: Interacciones entre genes y proteínas, y, las conexiones cerebrales.

Representar datos como grafos permite incorporar información estructural compleja como características. Existen diversas aplicaciones:

- Clasificación de nodos: Predecir la propiedad de un nodo.
- Predicción de relaciones: Predecir relaciones perdidas entre nodos.
- Clasificación de grafos: Categorizar diferentes grafos.
- Clustering: Detectar si los nodos forman parte de una comunidad.
- Generación de grafos: Investigación de drogas.
- Solución de grafos: Simulaciones físicas.

Adicionalmente estas estructuras permiten implementar técnicas de clasificación, regresión y clustering. Los algoritmos pueden ïncrustarçada nodo de un grafo en una estructura vectorial (de forma similar a la incrustación de una palabra). El resultado será una representación vectorial de cada nodo del grafo con cierta información conservada. Una vez que se tiene el vector de números reales, se puede implementar cualquier técnica de las previamente mencionadas.

5.1.2. Definición de Grafo

Un **grafo simple no dirigido** (o simplemente, un grafo) G está definido como una pareja $G = (v, E)$, donde $V = V_1, \ldots, V_n$ es un conjunto de nodos (A veces llamados vertices) y $E = V_k, V_w, \ldots, V_i, V_j$ es un conjunto de conjuntos de dos elementos de aristas (A veces llamados enlaces), que representan la conexión entre dos nodos pertenecientes a V.

Es importante mencionar que como cada elemento de E es un conjunto de dos elementos, no existe un orden entre las aristas, es decir, V_k, V_w y V_w, V_k representan la misma arista. Ahora se proporcionan las definiciones de las propiedades básicas de los nodos y de las aristas:

- El **orden** de un grafo es el número de nodos $|V|$. El **tamaño** de un grafo es el número de aristas $|E|$.
- El **grado** de un nodo es el número de aristas que son adyacentes a el. Los **vecinos** de un nodo v en un grafo G corresponden a un subconjunto de nodos de V inducido por todos los nodos adyacentes a v.
- El **grafo de vecindario** de un nodo v en un grafo G es un subgrafo de G compuesto por todos los nodos adyacentes a v y todas las aristas que conectan los nodos adyacentes a v.

5.1.3. Tipos de Grafos

Ahora bien, podemos extender las definiciones anteriores para introducir distintos tipos de grafos, dirigidos, múltiples y ponderados.

- **Grafos Dirigidos**

  Un grafo dirigido G está definido como una pareja $G = (v, E)$, donde $V = V_1, \ldots, V_n$ es un conjunto de nodos (A veces llamados vértices) y $E = V_k, V_w, \ldots, V_i, V_j$ es un conjunto de parejas ordenadas que representan la conexión entre dos nodos pertenecientes a V.

  Como cada elemento de E es una pareja ordenada, da lugar a establecer la dirección de la conexión. La arista (V_k, V_w) significa que el nodo V_k va hacía V_w, caso contrario a la arista (V_w, V_k). El nodo de inicio V_k es conocido como la cabeza y el nodo de fin es conocido como la cola.

- **Multigrafos**

  Un multigrafo G está definido como una pareja $G = (v, E)$, donde V es un conjunto de nodos y E es un conjunto múltiple de parejas que representan la conexión entre dos nodos pertenecientes a V. Si las parejas están ordenadas entonces el multigrafo es llamado multigrafo dirigido, en caso contrario, multigrafo no dirigido.

- **Grafos Ponderados**

  Un grafo ponderado por la arista G está definido como $G = (V, E, w)$ donde V es un conjunto de nodos, E es un conjunto de aristas y $w : E \longrightarrow R$ es una función ponderada que asigna a cada arista $e \in E$ un peso expresado como un número real.

  Un grafo ponderado por el nodo G está definido como $G = (V, E, w)$ donde V es un conjunto de nodos, E es un conjunto de aristas y $w : V \longrightarrow R$ es una función ponderada que asigna a cada nodo $v \in V$ un peso expresado como un número real.

5.1.4. Visualización de grafos

Todo grafo puede representarse de dos formas, con la matriz de adyacencia o con el listado de aristas.

- **Matriz de Adyacencia**

  La matriz de adyacencia M de un grafo $G = (V, E)$ es una matriz cuadrada en donde su elemento M_{ij} es 1 cuando hay una arista del nodo i al nodo j, y 0 cuando no existe dicha arista. Esta matriz siempre es simétrica para grafos no dirigidos, sin embargo esto no siempre se cumple para grafos dirigidos. Para el caso de los múltigrafos, podemos tener valores mayores a 1 puesto que puede haber más de una arista entre la misma pareja de nodos, y, para los grafos dirigidos el valor de la matriz corresponde al peso de la arista que conecta los dos nodos.

- **Lista de Aristas**

  La lista de aristas L de un grafo $G = (V, E)$ es una lista de tamaño $|E|$ donde cada elemento L_i corresponde a una pareja que representa la cabeza y la cola de la relación i.

5.1.5. Propiedades de grafos

Cada grafo presenta una serie de propiedades intrínsecas, las cuales pueden ser medidas por algunas métricas en particular, cada medida puede caracterizar uno o muchos aspectos locales y globales del grafo. Anteriormente vimos que el número de nodos y de aristas de un grafo constituyen por si solos el tamaño de un grafo, estas propiedades proveen una buena descripción de la estructura de una red, sin embargo, pueden no ser suficientes para caracterizar estructuras más complejas. Para este fin, una serie de métricas más avanzadas pueden ser consideradas, las cuales se agrupan en 4 categorías principales, integración, segregación, centralidad y resiliencia.

Estas métricas son consideradas **globales** cuando expresan una medida sobre toda la red. Por otro lado, las métricas **locales** miden valores de elementos individuales de la red. En los grafos **ponderados**, cada propiedad puede tener en cuenta o no los pesos de las aristas, lo que da lugar a métricas ponderadas y no ponderadas.

Métricas de Integración

Estás métricas tienen miden cómo tienden a interconectarse los nodos entre sí.

El concepto de **distancia** de un grafo suele estar relacionado con el número de aristas que hay que recorrer para llegar a un nodo destino desde un nodo origen. En particular, considere un nodo origen i y un nodo destino j. El conjunto de aristas que conectan el nodo i al nodo j es llamado **ruta**. Cuando se estudian redes complejas, se suele estar interesado en hallar la **ruta más corta** entre dos nodos. La ruta más corta entre el nodo origen i y el nodo destino j es la ruta con el menor número de aristas comparado con todas las posibles rutas entre i y j. El **diámetro** de una red es el número de aristas contenidas en la ruta más corta entre todas las rutas más cortas posibles. A continuación se presentan algunas de las métricas:

- **Characteristic Path Length**

  La **longitud de ruta característica** se define como la media de todas las longitudes de ruta más cortas entre todos los pares de nodos posibles. Si l_i es la longitud media de la ruta entre el nodo i y todos los demás nodos, la longitud de ruta característica se calcula de la siguiente manera:

  $$\frac{1}{q(q-1)} \sum_{i \in V} l_i$$

  Donde V es el conjunto de nodos en el grafo y $q = |V|$ representa el **orden**. Esta es una de las métricas más utilizadas para medir la eficacia con la que se difunde la información a través de una red. Las redes con longitudes de ruta características más cortas favorecen la transferencia rápida de información y disminuyen los costos. Sin embargo, está métrica no siempre se puede definir ya que en algunos casos no es posible calcular la ruta entre todos los nodos en un **grafo desconectado**. Por esta razón, se suele usar la **eficiencia** de la red como métrica.

- **Global and Local Efficiency**

  La **eficiencia global** es la media de la longitud inversa de la ruta más corta para todos los pares de nodos. Esta métrica puede considerarse una medida de la eficacia con la que se intercambia información a través de la red. Considere que l_{ij} es la ruta más corta entre los nodos i y j. La eficiencia está definida de la siguiente manera:

  $$\frac{1}{q(q-1)} \sum_{i \in V} \frac{1}{l_{ij}}$$

  La eficiencia es máxima cuando un grafo está completamente conectado, mientras que es mínima para grafos completamente desconectados. Intuitivamente, cuando más corta es la ruta, menor es la medida.

  La **eficiencia local** de un nodo se puede calcular teniendo en cuenta sólo el vecindario del nodo en el cálculo, sin el propio nodo.

Las métricas de **integración** describen bien la conexión entre nodos. Sin embargo, se puede extraer más información sobre la presencia de grupos considerando las métricas de segregación.

Métricas de Segregación

Cuantifican la presencia de grupos de nodos interconectados, conocidos como comunidades o módulos dentro de una red. A continuación se presentan algunas de estas métricas:

- **Clustering Coefficient**

  Es una medida del grado de agrupación de los nodos. Se define como la fracción de **triángulos** (subgrafo completo de tres nodos y tres relaciones) alrededor de un nodo y equivale a la fracción de vecinos del nodo que son vecinos entre sí.

- **Transitivity**

  Se puede definir simplemente como la relación entre el número observado de **tripletas cerradas** (subgrafo completo con tres nodos y dos relaciones) y el número máximo posible de tripletas cerradas en el grafo.

- **Modularity**

  Se diseñó para cuantificar la división de una red en conjuntos agregados de nodos altamente interconectados, conocidos comúnmente como módulos, comunidades, grupos o clusters. La idea principal es que las redes con alta modularidad mostrarán conexiones densas dentro del módulo y conexiones dispersas entre módulos. A diferencia de otras métricas, la modularidad suele calcularse mediante algoritmos de optimización.

Las métricas de segregación ayudan a comprender la presencia de grupos. Sin embargo, cada nodo de un grafo tiene su propia importancia. Para cuantificarla, podemos utilizar métricas de centralidad.

Métricas de Centralidad

Tienen como objetivo evaluar la importancia de los nodos individuales dentro de una red. A continuación se presentan algunas de estas métricas:

- **Degree Centrality**

  Está directamente relacionado con el grado de un nodo y mide el número de aristas incidentes en un determinado nodo.

  Intuitivamente, cuanto más conectado esté un nodo a otro, más alta será su centralidad de grado. Observe que, si un grafo es dirigido, para cada nodo se considerará la centralidad de grado de entrada y la centralidad de grado de salida, relacionadas con el número de aristas entrantes y salientes, respectivamente.

- **Closeness Centrality**

  Esta métrica intenta cuantificar hasta qué punto un nodo está cerca (bien conectado) de otros nodos. Más formalmente, se refiere a la distancia media de un nodo a todos los demás nodos de la red. Si l_{ij}es la ruta más corta entre el nodo i y el nodo j, la métrica se define de la siguiente manera:

  $$\frac{1}{\sum_{i \in V, i!=j} l_{ij}}$$

  Donde V es el conjunto de nodos en el grafo.

- **Betweenness Centrality**

  Esta métrica evalúa en que medida un nodo actúa como puente entre otros nodos. Aunque esté mal conectado, un nodo puede estar estratégicamente conectado ayudando a mantener conectada a toda la red.

  Si L_{wj} es el número total de rutas más cortas entre el nodo w y el nodo j y $L_{wj}(i)$ es el número total de rutas más cortas entre w y j que pasan por el nodo i, entonces la métrica se define de

la siguiente manera:

$$\sum_{w!=i!=j} \frac{L_{wj}(i)}{L_{wj}}$$

Al observar la fórmula se identifica que cuanto mayor sea el número de rutas más cortas que pasan por el nodo i, mayor sera el valor de la métrica.

Las métricas de centralidad nos permiten medir la importancia de un nodo dentro de la red. Por último, las métricas de resiliencia nos permiten medir la vulnerabilidad de un grafo.

Métricas de Resiliencia

Las métricas de resilencia se pueden considerar como una medida del grado en que una red es capaz de mantener y adaptar su rendimiento operativo cuando se enfrenta a fallos u otras condiciones adversas.

Existen múltiples métricas, la **asortatividad** es una de las más usadas, está métrica se utiliza para cuantificar la tendencia de los nodos a estar conectados a nodos similares.

Hay varias formas de medir estas correlaciones, uno de los métodos más utilizados es el **coeficiente de correlación de Pearson** entre los grados de los nodos conectados directamente (nodos situados en dos extremos opuestos de una arista). El coeficiente asume valores positivos cuando existe una correlación entre nodos de grado similar, mientras que asume valores negativos cuando existe una correlación entre nodos de un grado diferente.

Las métricas presentadas anteriormente corresponden a un subconjunto de todas las posibles métricas usadas para describir grafos. El conjunto completo de métricas y algoritmos puede ser encontrada en la documentación de la librería **networkx**.

Toda la información presentada en esta sección del documento es tomada de [15].

5.2. Graph Machine Learning

5.2.1. Machine Learning

El Machine Learning, también conocido como aprendizaje automático, corresponde a una clase de algoritmos informáticos que aprenden y mejoran automáticamente sus habilidades a través de la experiencia sin ser programados explícitamente. El objetivo es encontrar un modelo matemático capaz de lograr el mejor rendimiento posible en una tarea concreta. El rendimiento puede medirse utilizando una métrica de rendimiento específica (también conocida como función de perdida o función de coste).

En una tarea de aprendizaje común, el algoritmo recibe datos, luego los utiliza para tomar decisiones o realizar predicciones de forma iterativa para la tarea específica. En cada iteración, las decisiones se evalúan utilizando la función de perdida. El error resultante se utiliza para actualizar los parámetros del modelo de forma que funcione mejor. Este proceso suele denominarse **entrenamiento**.

El aprendizaje automático se divide en tres categorías, conocidas como aprendizaje supervisado, no supervisado y semi supervisado. Estos paradigmas de programación dependen de la forma en la que se le proporcionan los datos al algoritmo y de cómo se evalúa su rendimiento.

Aprendizaje Supervisado

Es el paradigma utilizado cuando conocemos la respuesta al problema. En este escenario, el conjunto de datos se compone de muestras de pares de la forma (x, y), donde x es la entrada y y es la salida deseada correspondiente. Las variables de entrada también se conocen como características, mientras

que la salida suele denominarse etiquetas u objetivos. En entornos supervisados, el rendimiento suele evaluarse mediante una función de distancia, esta función mide las diferencias entre la predicción y el resultado esperado. Según el tipo de etiquetas, el aprendizaje supervisado puede dividirse en los siguientes casos:

- **Clasificación:** En este caso las etiquetas son discretas y se refieren a la çlase.ª la que pertenece la entrada. Aplicaciones: Determinar el objeto de una foto o predecir si un correo electrónico es spam o no.
- **Regresión:** El objetivo es continuo. Aplicaciones: Predicción de temperatura de un edificio o la predicción del precio de venta de un determinado producto.

Aprendizaje No Supervisado

A diferencia del aprendizaje supervisado en este caso no se conoce la respuesta al problema. En este contexto, no disponemos de etiquetas y sólo se proporcionan las entradas x. Por lo cual, el objetivo es deducir estructuras y patrones, intentando encontrar similitudes.

Descubrir grupos de ejemplos similares (Clustering) es uno de estos problemas así como dar nuevas representaciones de los datos de un espacio de alta dimensionalidad.

Aprendizaje Semi Supervisado

En este caso, el algoritmo se entrena utilizando una combinación de datos etiquetados y no etiquetados. Normalmente, para dirigir la investigación en los datos de entrada no etiquetados, se utiliza una cantidad limitada de datos etiquetados.

Ahora bien, no basta con minimizar el error en los datos de entrenamiento, los algoritmos deben ser capaces de alcanzar el mismo nivel de rendimiento incluso con datos no vistos. La forma más habitual de evaluar la capacidad de generalización de estos algoritmos consiste en dividir el conjunto de datos en dos partes, el **conjunto de entrenamiento** y el **conjunto de prueba**. El modelo se entrena en el conjunto de entrenamiento, donde se calcula la función de perdida y se utiliza para actualizar los parámetros. Tras el entrenamiento, se evalúa el rendimiento del modelo en el conjunto de prueba.

Cuando se entrena un algoritmo de aprendizaje automático, se pueden observar tres situaciones:

- **Underfitting:** El modelo alcanza un bajo nivel de rendimiento sobre el conjunto de entrenamiento, por lo cual, se concluye que el modelo no es lo suficientemente potente para abordar la tarea.
- **Overfitting:** El modelo alcanza un alto nivel de rendimiento en el conjunto de entrenamiento, pero tiene dificultades para generalizar en los datos de prueba. En este caso, ek modelo se limita a memorizar los datos de entrenamiento, sin comprender realmente las verdaderas relaciones entre ellas.
- La situación ideal en la que el modelo es capaz de alcanzar el máximo nivel de rendimiento tanto en los datos de entrenamiento como en los de prueba.

Se han desarrollado múltiples algoritmos de aprendizaje automático, cada uno con sus propias ventajas y limitaciones. Por ejemplo, algoritmos de regresión (Regresión Lineal y Logística), algoritmos basados en instancias (K Nearest Neighbor o Máquinas de Soporte Vectorial), algoritmos de árboles de decisión, algoritmos Bayesianos (Naive Bayes), algoritmos de agrupación (K-Means) y redes neuronales artificiales.

Ahora bien, el aprendizaje automático de grafos permite crear algoritmos para detectar e implementar automáticamente patrones latentes recurrentes.

5.2.2. Graph Machine Learning

Debido a su naturaleza, los grafos pueden analizarse a distintos niveles de granularidad: A nivel de nodo, arista y grafo (todo el grafo).

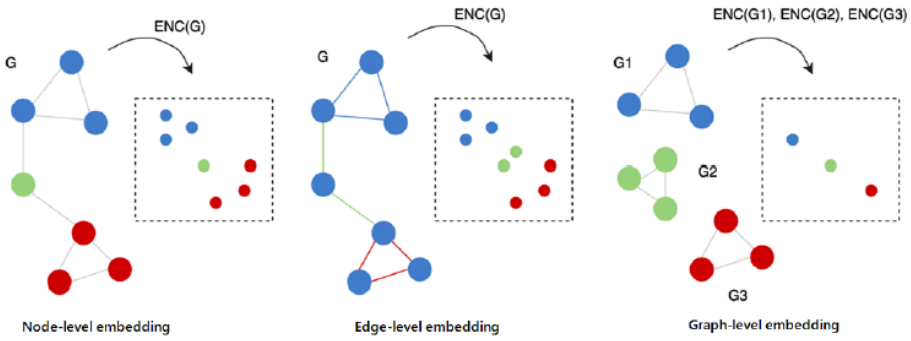

Figura 1: Representación de los tres niveles de granularidad en grafos

Imagen tomada de [15]. Para cada uno de esos niveles pueden plantearse distintos problemas y, en consecuencia, deben utilizarse algoritmos específicos. A continuación, se presentan algunos ejemplos:

- **Nivel de nodo:** Dado un grafo, $G = (V, E)$, el objetivo es clasificar cada nodo, $v \in V$, en la clase correcta. En este caso, el conjunto de datos incluye G y una lista de parejas (v_i, y_i), donde v_i es un nodo del grafo G y y_i es la clase a la que pertenece el nodo.
- **Nivel de arista:** Dado un grafo, $G = (V, E)$, el objetivo es clasificar cada arista, $e \in E$, en la case correcta. En este caso, el conjunto de datos incluye G y una lista de parejas (e_i, y_i), donde e_i es una arista del grafo G y y_i es la clase a la que pertenece la arista. Otra aplicación común sobre este nivel de granularidad es la predicción de aristas.
- **Nivel de grafo:** Dado un conjunto de datos con m grafos diferentes, la tarea es construir un algoritmo de aprendizaje automático capaz de clasificar a los nodos en la clase correcta. En este caso, el conjunto de datos es una lista de parejas (G_i, y_i), donde G_i es un grafo G y y_i es la clase a la que pertenece el grafo.

En las aplicaciones clásicas de aprendizaje automático, una forma habitual de procesar los datos de entrada es construir a partir de un conjunto de características, en un proceso llamado **Feature Engineering**, el cual es capaz de ofrecer una representación compacta y significativa de cada variable presente en el conjunto de datos. El resultado de este ejercicio se utiliza como entrada para el algoritmo de aprendizaje automático.

Este proceso suele funcionar para una amplía gama de problemas, sin embargo puede no ser la solución óptima cuando se trata de grafos. Dada su estructura bien definida, encontrar una representación adecuada capaz de incorporar toda la información útil puede no ser tarea fácil.

La primera forma de crear características capaces de representar la información estructural de los grafos es la extracción de determinados estadísticos, por ejemplo, un grafo puede representarse mediante su distribución de grados, su eficiencia y todas las métricas presentadas anteriormente. Un procedimiento más complejo consiste en aplicar funciones kernel específicas que sean capaces de incorporar las propiedades deseadas al modelo final de aprendizaje automático, sin embargo, esto puede conllevar mucho tiempo, y, en algunos casos, las características utilizadas pueden representar sólo un subconjunto de la información que realmente se necesita para obtener el mejor rendimiento posible.

Ahora bien, en los últimos años se han desarrollado nuevos enfoques llamados **Representation Learning** o **Network Embedding**, cuya idea general es crear algoritmos capaces de aprender una

buena representación del conjunto de datos original, de forma que las relaciones geométricas en el nuevo espacio reflejen la estructura del grafo original.

Representation Learning

Es la tarea que tiene como objetivo aprender una función de mapeo, $f : G \longrightarrow R^n$, de un grafo discreto a un dominio continuo. La función debe ser capaz de realizar una representación vectorial de baja dimensión de forma que se preserven las propiedades (locales y globales) del grafo.

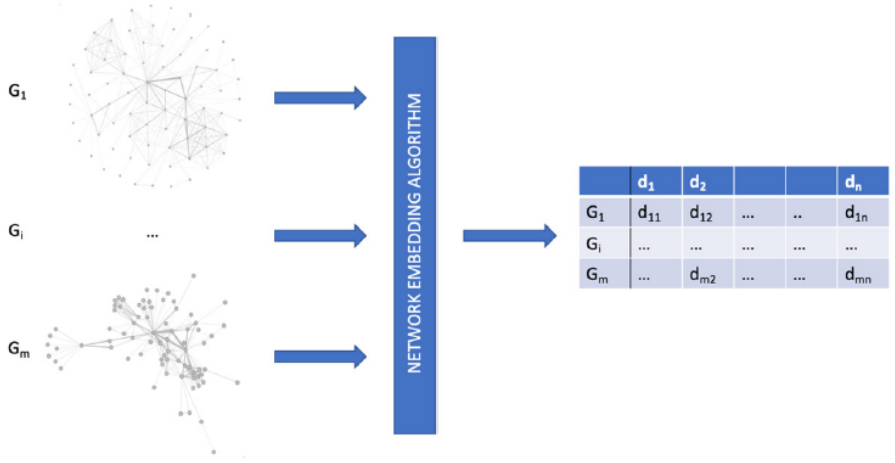

Figura 2: Representación de un algoritmo de Network Embedding

Una vez aprendido el mapeo, puede aplicarse al grafo y el mapeo resultante puede usarse como conjunto de características para un algoritmo de aprendizaje automático. La función de mapeo también puede aplicarse para aprender la representación vectorial de nodos $f : V \longrightarrow R^n$ conocida como **node embedding** o aristas $f : E \longrightarrow R^n$ conocida como **edge embedding**. Estas funciones de mapeo intentan construir un espacio vectorial tal que las relaciones geométricas en el nuevo espacio reflejen la estructura del grafo, nodo o arista originales. En otras palabras, en el espacio generado por la función de embbeding, las estructuras similares tendrán una distancia euclidiana pequeña, mientras que las estructuras diferentes tendrán una distancia euclidiana grande.

Estos algoritmos de Embedding pueden clasificarse en cuatro grupos principales, Shallow Embedding Methods, Graph Autoencoding Methods, Neighborhood Aggregation Methods y Graph Regularization Methods. En este trabajo nos centraremos en el primer grupo.

Shallow Embedding Methods

Estos métodos son capaces de aprender y devolver sólo los valores de embbeding para los datos de entrada aprendidos. **Node2Vec**, **Edge2Vec** y **Graph2Vec** son ejemplos de este tipo de métodos, estos métodos no permiten obtener el vector de embeddings para datos no vistos. Para estos algoritmos existe la posibilidad de definir una versión supervisada y otra no supervisada.

En el caso de los algoritmos no supervisados de embeddings de grafos, dado un grafo, el objetivo de estás técnicas es aprender automáticamente una representación latente del mismo, en la que se preserven de algún modo los componentes estructurales claves. En este trabajo nos centraremos en el algoritmo **Node2Vec**.

Toda la información presentada en esta sección del documento es tomada de [15].

5.2.3. Algoritmo Node2Vec

El algoritmo **Node2Vec** hace parte de un subconjunto de técnicas basadas en el modelo **Skip-gram**, por lo cual se requiere entender su funcionamiento.

El modelo **Skip-gram** es una red neuronal simple con una capa oculta entrenada para predecir la probabilidad de que la palabra dada esté presente cuando una palabra de entrada está presente. La red neuronal se entrena construyendo los datos de entrenamiento utilizando un corpus de texto como referencia. El proceso se describe en la siguiente imagen:

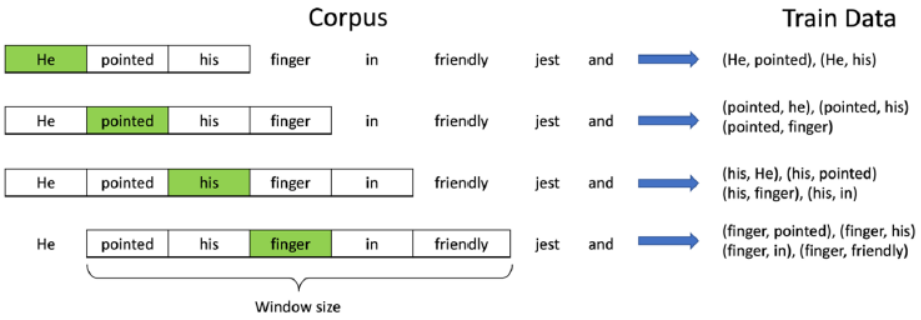

Figura 3: Ejemplo de la generación de datos de entrenamiento a partir de un corpus dado

Se selecciona una palabra objetivo y se construye una ventana móvil de tamaño fijo w alrededor de esa palabra. Las palabras dentro de las ventanas móviles se conocen como palabras de contexto. A continuación se crean varios pares de (palabra objetivo, palabra de contexto) en función de las palabras de la ventana móvil.

Una vez generados los datos de entrenamiento a partir de todo el corpus, se entrena el modelo Skip-gram para predecir la probabilidad de que una palabra sea una palabra de contexto para el objetivo dado. Durante el entrenamiento, la red neuronal aprende una representación compacta de las palabras de entrada.

La estructura de la red neuronal que representa el modelo Skip-gram se describe en el siguiente gráfico:

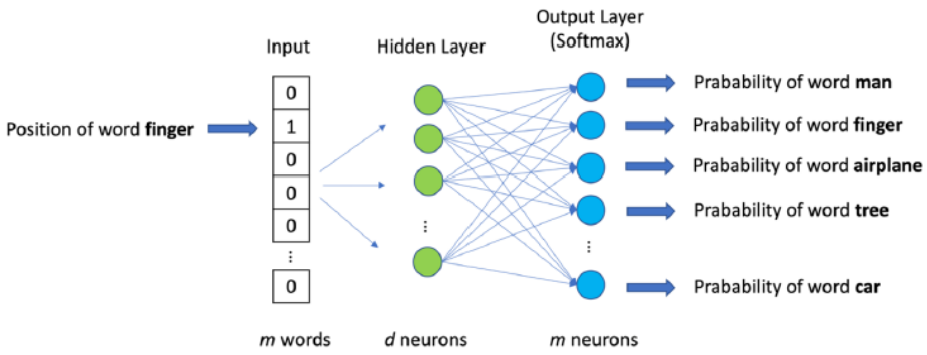

Figura 4: Estructura de la red neuroanl del modelo Skip-Gram

La entrada de la red neuronal es un vector binario de tamaño m. Cada elemento del vector representa una palabra del diccionario de la lengua en la que queremos incrustar las palabras. Cuando, durante el proceso de entrenamiento, se da un par (palabra objetivo, palabra de contexto), la matriz de entrada tendrá 0 en todas sus entradas a excepción de la entrada que representa la palabra

"objetivo", que será igual a 1. La capa oculta tiene d neuronas y aprenderá la representación de incrustación de cada palabra, creando un espacio de incrustación $d-$dimensional. Por último, la capa de salida de la red neuronal es una capa densa de $m-$neuronas (del mismo tamaño que el vector de entrada) con una función de activación softmax. El valor asignado por la neurona corresponde a la probabilidad de que esa palabra esté relacionadaçon la palabra de entrada. Dado que Softmax puede ser díficil de calcular cuando aumenta el tamaño de m, siempre se utiliza un enfoque softmax jerárquico.

El objetivo final del modelo Skip-gram no es aprender la tarea descrita anteriormente, sino construir una representación d-dimensional compacta de las palabras de entrada. Gracias a esta representación, es posible extraer fácilmente un espacio de embbedings para las palabras utilizando el peso de la capa oculta.

A partir de un grafo de entrada, extraen de él un conjunto de caminatas. Esas caminatas pueden verse como un corpus de texto en el que cada nodo representa una palabra. Dos palabras (que representan nodos) están cerca una de otra en el texto si están conectadas por una arista en una caminata.

Ahora necesitamos introducir el concepto de **random walks**. Sea G un grafo y sea v_i un nodo seleccionado como punto de partida. Seleccionamos un vecino al azar y nos movemos hacía él. Desde ese punto, seleccionamos al azar otro punto para movernos. Este proceso se repite t veces. La secuencia aleatoria de nodos seleccionados de esta manera es un paseo aleatorio de longitud t.

El algoritmo **Node2Vec** genera un conjunto de caminatas aleatorias que se utilizan como entrada para el modelo Skip-gram, una vez entrenadas las capas ocultas del modelo se utilizan para generar el embbeding del nodo en el grafo.

El algoritmo para generar las caminatas aleatorias combina la exploración de grafos mediante la fusión de los algoritmos **Breadth-First-Search** (BFS) y **Depth-First-Search** (DFS). La forma en la que estos dos algoritmos se combinan se regulariza mediante dos parámetros, p y q, p define la probabilidad de que una camina aleatoria vuelva al nodo anterior, mientras que q define la probabilidad de que una camina aleatoria pueda pasar por una parte del grafo que no se haya explorado antes. Gracias a esta combinación, **Node2Vec** puede preservar las proximidades de alto orden conservando las estructuras locales del grafo, así como las estructuras comunitarias globales.

Ya que se conoce el funcionamiento del modelo Skip-Gram y el algoritmo encargado de generar las caminatas aleatorias, se procede a presentar una explicación del algoritmo **Node2Vec**:

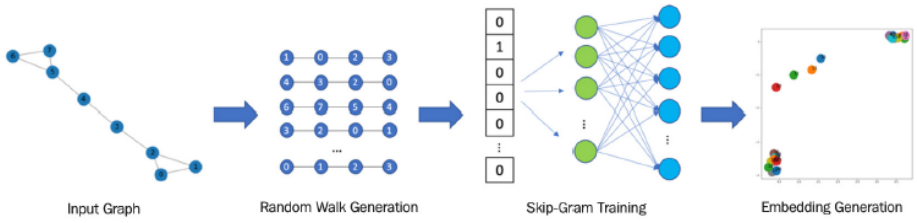

Figura 5: Algoritmo Node2Vec

1. **Random Walk Generation:** Para cada nodo de entrada del grafo G, se calcula un conjunto de caminatas aleatorias con una longitud máxima fija t. Cabe señalar que la longitud t es un límite superior.
2. **Skip-Gram Training:** Utilizando todas las caminatas aleatorias generadas en el paso anterior, se entrena un modelo Skip-Gram. Cuando se da un grafo como entrada al modelo, un grafo puede verse como un corpus de texto de entrada, mientras que un solo nodo del grafo puede verse como una palabra del corpus.

   Una caminata aleatoria puede verse como una secuencia de palabras (una frase), a continuación, se entrena el modelo Skip-Gram utilizando solo las frases "falsas"generadas por los nodos en

en recorrido aleatorio. En este punto se específican los parámetros del modelo, el tamaño de la ventana, w, y, el tamaño del embbeding, d.

3. **Embbeding Generation:** La información contenida en las capas ocultas del modelo Skip-Gram entrenado se utliza para extraer el emebbding de cada nodo.

El algoritmo **Node2Vec** acepta otros parámetros como **num-walks** que corresponde al número de caminatas aleatorias que se generan para cada nodo, **walk-length** que corresponde al tamaño de las caminatas aleatorias generadas y p y q que correspoden a los parámetros del algoritmo que genera las caminatas aleatorias.

Toda la información presentada en esta sección del documento es tomada de [15].

5.3. Detección de Anomalías

Las anomalías pueden definirse como observaciones que se desvían lo suficiente de la mayoría de las observaciones del conjunto de datos como para considerar que han sido generadas por un proceso generativo diferente, no normal. Una anomalía es cualquier observación que se desvía tanto de las demás observaciones del conjunto de datos como para despertar sospechas. En resumen, las anomalías son observaciones raras y significativamente diferentes dentro de un conjunto de datos.

Los algoritmos de detección de anomalías se utilizan actualmente en muchos ámbitos de aplicación para la detección de intrusiones, la detección de fraudes, la prevención de fugas de datos, la calidad de los datos y la vigilancia y el control. Como se puede ver, se trata de una gran variedad de aplicaciones, algunas de las cuales requieren una detección de anomalías muy rápida y casi en tiempo real, mientras que otras requieren un rendimiento muy elevado debido al alto coste que supone no detectar una anomalía. Las técnicas de detección de anomalías se utilizan más comúnmente para detectar el fraude, donde los intentos/transacciones maliciosas a menudo difieren de la mayoría de los casos nominales. A continuación, se describen los diferentes tipos de anomalías:

- Anomalía puntual: casos anómalos individuales en un conjunto de datos mayor.
- Anomalía colectiva: Si una situación anómala se representa como un conjunto de muchos casos, se denomina anomalía colectiva.
- Anomalía contextual: En las anomalías contextuales, el punto puede considerarse normal, pero cuando se tiene en cuenta un contexto determinado, el punto resulta ser una anomalía.

La solución a la detección de anomalías puede enmarcarse en los tres tipos de métodos de aprendizaje automático: supervisado, semi-supervisado y no supervisado, en función del tipo de datos disponibles. Los algoritmos de aprendizaje supervisado pueden utilizarse para la detección de anomalías cuando éstas ya se conocen y se dispone de datos etiquetados. Estos métodos son especialmente costosos cuando el etiquetado tiene que hacerse manualmente. Los algoritmos de clasificación no supervisados, como las máquinas de vectores de apoyo (SVM) o las redes neuronales artificiales (ANN), pueden utilizarse para la detección supervisada de anomalías.

La detección de anomalías semi-supervisada utiliza datos etiquetados que consisten únicamente en datos normales sin anomalías. La idea básica es que se aprende un modelo de la clase normal y cualquier desviación de ese modelo puede considerarse una anomalía. Algoritmos populares: Auto-Encodificadores, Modelos de Mezcla Gaussiana, Estimación de Densidad Kernel.

Los métodos de aprendizaje no supervisado son los más utilizados para detectar anomalías, el siguiente cuadro resume las principales familias de algoritmos y los algoritmos que pueden utilizarse para la detección de anomalías.

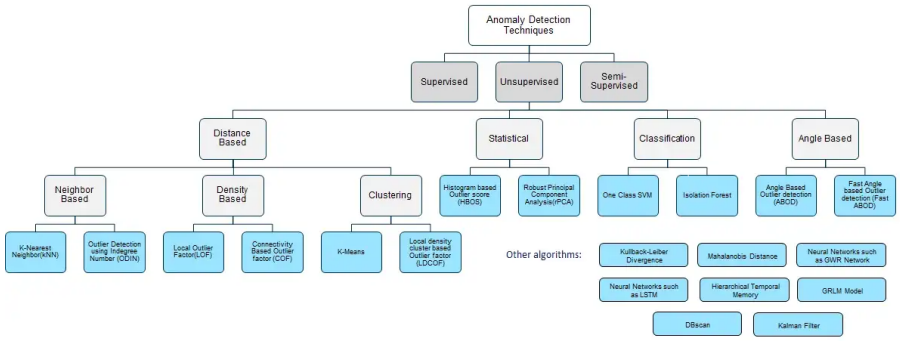

Figura 6: Algoritmos Detección Anomalías

La mayoría de los enfoques basados en modelos para la detección de anomalías construyen un perfil de observaciones normales, y luego identifican las observaciones que no se ajustan al perfil normal como anomalías. Ejemplos notables como los métodos estadísticos, los métodos basados en clasificación y los métodos basados en la agrupación utilizan este enfoque general. Dos inconvenientes importantes de este enfoque son: (Tomado de [16])

1. El detector de anomalías está optimizado para perfilar observaciones normales, pero no está optimizado para detectar anomalías, por lo que los resultados de la detección de anomalías pueden no ser los esperados.

2. Muchos métodos existentes están limitados a datos de baja dimensión y a un tamaño de datos pequeño debido a su alta complejidad computacional.

5.4. Isolation Forest

El método de bosques de aislamiento (Isolation Forest) se caracteriza por aislar explícitamente las anomalías en lugar de perfilar las observaciones normales. Este método aprovecha dos propiedades cuantitativas de las anomalías, a) son la minoría compuesta por menos observaciones y b) tienen valores de atributos muy diferentes a los de las observaciones normales.

Es decir, las anomalías son pocas y diferentes, lo que las hace más susceptibles de aislamiento que los puntos normales. Debido a su susceptibilidad al aislamiento las anomalías se aíslan más cerca de la raíz del árbol, mientras que los puntos normales se aíslan en el extremo más profundo del árbol. Esta característica de aislamiento constituye la base del método para detectar anomalías, y llamamos a este árbol **Isolation Tree** o **iTree**.

El método Isolation Forest construye un conjunto de iTrees para un conjunto de datos dado, y entonces las anomalías son aquellas observaciones que tienen una profundidad corta en los iTrees. Solo hay dos variables en este método: el número de árboles a construir y el tamaño del submuestreo.

Ahora bien, el término aislamiento significa "separar una observación del resto de las observaciones". Dado que las anomalías son pocas y diferentes y, por tanto, son más susceptibles al aislamiento. En el árbol aleatorio inducido por datos, la partición de las observaciones se repite recursivamente hasta que todas las observaciones están aisladas. Esta partición aleatoria produce caminos notablemente más cortos para las anomalías, ya que:

- El menor número de observaciones de anomalías da lugar a un menor número de particiones (Caminos más cortos en una estructura de árbol).

- Las observaciones con valores de atributos distinguibles tienen más probabilidad de ser separadas en la primera partición.

Por lo tanto, cuando un bosque de árboles aleatorios produce colectivamente trayectorias más cortas para algunos puntos particulares, entonces es muy probable que sean anomalías.

Definición: Árboles de aislamiento. Dado un conjunto de datos d−variados $X = \{X_1, \ldots, X_n\}$. Para construir un árbol de aislamiento se divide X seleccionando de manera aleatoria una variable q y un valor de separación p, hasta que: (i) el árbol alancé su altura máxima, o (ii) $|x| = 1$, o (iii) todos los datos en X tengan el mismo valor. Cada nodo de este árbol tendrá únicamente dos o cero hijos. Suponiendo que todas las instancias son distintas, cada observación se aísla en un nodo externo cuando un iTree crece por completo, en cuyo caso el número de nodos externos es n y el número de nodos internos es n - 1; el número total de nodos de un iTrees es 2n - 1; y, por tanto, el requisito de memoria está acotado y sólo crece linealmente con n.

La tarea de la detección de anomalías es proporcionar una clasificación que refleje el grado de anomalía. Por lo tanto, una forma de detectar anomalías es ordenar los puntos de datos según la longitud de sus trayectorias o la puntuación de las anomalías; y las anomalías son puntos que se sitúan en la parte superior de la lista. Se define la profundidad y la puntuación de la anomalía de la siguiente manera.

Definición: Profundidad Se define $h(x)$ como la cantidad de nodos que debe recorrer una observación x dentro de un árbol de aislamiento para llegar a un nodo terminal, es decir, quedar aislado.

Definición: Puntuación Dado un conjunto de datos con n observaciones se define la longitud media del camino de la búsqueda infructuosa de la siguiente manera

$$c(n) = 2H(n-1) - 2\frac{(n-1)}{n}$$

donde $H(i)$ es el número armónico, que se puede estimar como $H(i) = ln(i) + e$, la constante de Euller. Como $c(n)$ es la media de $h(x)$ dado n, se utiliza para normalizar $h(x)$. Luego, la puntuación de anomalía s de una instancia x se define como:

$$s(x, n) = 2^{\frac{E(h(x))}{c(n)}}$$

donde $E(h(x))$ es el promedio de $h(x)$ sobre una colección de arboles de aislamiento. A partir del puntaje de anomalía s se puede realizar la siguiente evaluación: (Tomado de [4])

- Si el puntaje s de una observación es muy cercano a 1, definitivamente es una anomalía.
- Si el puntaje s de una observación es mucho menor a 0.5, es una valor normal.
- Si el puntaje s de todas las observaciones es aproximadamente igual a 0.5, no existen anomalías.

5.5. Histogram Based Outlier Score

El algoritmo HBOS permite aplicar la detección de anomalías basada en histogramas de forma general. Para cada característica individual (dimensión) se construye primero un histograma univariado. Si la característica consta de datos categóricos, se realiza un simple recuento de los valores de cada categoría y se calcula la frecuencia relativa (Altura del histograma). Para las características numéricas, se pueden utilizar dos métodos diferentes:

- Histogramas de anchura estática.
- Histogramas de anchura dinámica.

El primero es la técnica estándar de construcción de histogramas, que utiliza k bloques de igual anchura en el rango de valores. La frecuencia (cantidad relativa) de las muestras que caen en cada bloque se utiliza como una estimación de la densidad (altura de los intervalos). Por otra parte, la anchura dinámica se determina de la siguiente manera: los valores se ordenan primero y, a continuación, una cantidad determinada de $\frac{N}{k}$ valores sucesivos se agrupan en un único bloque, donde N es el número de total de observaciones y k el número de bloques. Dado que el área de un bloque en un histograma representa el número de observaciones, es igual para nuestro caso. Dado que la anchura del bloque se

determina por el primer y el último valor, y, que el área es la misma para todos los bloques, se puede calcular la altura de cada uno de ellos.

Esto significa que los bloques que cubren un mayor intervalo del rango de valores tienen menor altura y por ende, representan una menor densidad. Sin embargo, hay una excepción, en determinadas circunstancias, más de k observaciones pueden tener exactamente el mismo valor, por ejemplo, si la característica es un número entero y hay que estimar una distribución de cola larga. En este caso, el algoritmo debe permitir que haya más de $\frac{N}{k}$ valores en el mismo bloque. Por supuesto, el área de estos bloques más grandes crecerá adecuadamente.

La razón por la que se utilizan ambos métodos en HBOS es el hecho de que hay distribuciones muy diferentes de las observaciones en el mundo real. Especialmente cuando los rangos de valores tienen grandes huecos (Bloques sin observaciones), el enfoque de anchura estática estima la densidad de manera pobre (Unos pocos bloques pueden contener la mayoría de los datos). Dado que las tareas de detección de anomalías suelen implicar estos huecos en los rangos de valores debido a que los valores atípicos están muy alejados de los datos normales, se recomienda utilizar el enfoque de anchura dinámica, especialmente si las distribuciones son desconocidas o de colas pesadas. Además, también es necesario establecer el número de bloques k. Una regla general que se utiliza con frecuencia es fijar k en la raíz cuadrada del número de observaciones N. Ahora, para cada dimensión se ha calculado un histograma univariado (independientemente de si es categórico, de anchura fija o de anchura dinámica), donde la altura de cada bloque individual representa la estimación de la densidad. A continuación, los histogramas se normalizan de forma que la altura máxima sea 1. Esto garantiza que cada característica tenga el mismo peso en la puntuación de los valores atípicos.

Por último, el HBOS de cada observación p se calcula utilizando la altura correspondiente de los bloques donde se encuentra la respectiva observación.

$$HBOS(p) = \sum_{i=1}^{d} log(\frac{1}{hist_i(p)})$$

El score es la multiplicación de la inversa de las densidades estimadas asumiendo independencia entre las característias (variables) de las observaciones. Esto también podría verse como (la inversa de) un modelo de probabilidad discreta de Naive Bayes. En lugar de la multiplicación, tomamos la suma de los logaritmos, que es básicamente lo mismo ($log(a*b) = log(a)+log(b)$) y aplicando $log(*)$ no cambia el orden de los scores.

Se puede observar que el algoritmo es bastante sencillo, aún así, es un algoritmo con un buen desempeño, a continuación se presenta una serie de sus ventajas:

1. Es un algoritmo eficiente en términos de ejecución, especialmente en conjuntos grandes de datos. Simplemente utiliza la estructura del histograma para calcular los puntajes de anomalía de manera rápida. Funciona en tiempo lineal $O(n)$ en caso de anchura estática o en $O(n - Log(n))$ en caso de anchura dinámica.

2. Es un algoritmo relativamente fácil de implementar y no requiere una configuración compleja de parámetros, solo necesita el número de intervalos del histograma.

3. Puede manejar conjuntos de datos de alta dimensionalidad, razón por la cual es un algoritmo escalable.

4. Es un algoritmo tolerante al ruido y a los valores atípicos, estos últimos generalmente tendrán scores de anomalías más altos.

Sin embargo, también presenta una serie de desventajas:

1. Este algoritmo asume que las características de las observaciones son independientes y que las distribuciones son unimodales y parecidas, lo cual no necesariamente se presenta en todos los conjuntos de datos.

2. Es un algoritmo sensible a la elección del número de intervalos, el cual puede influir en la precisión para detectar anomalías.

3. Presenta dificultades para detectar anomalías en regiones de baja densidad en los datos, razón por la cual puede ser menos efectivo en conjuntos de datos donde las anomalías están dispersas en regiones poco densas.

En resumen, HBOS modela densidades de características univariadas utilizando histogramas con un ancho de intervalo estático o dinámico. Posteriormente todos los histogramas se utilizan para calcular una puntuación de anomalía para cada instancia de datos. Tomado de [17].

5.6. Principal Component Analysis

El análisis de componentes principales es un algoritmo ampliamente utilizado en el campo de la estadística y el aprendizaje automático para reducir la dimensionalidad de conjuntos de datos. PCA se utiliza para encontrar una representación más compacta y significativa de los datos al proyectarlos en un nuevo espacio de menor dimensión. A continuación, se presenta una descripción detallada del algoritmo PCA:

- **Cálculo de la matriz de covarianza:** El primer paso en PCA es calcular la matriz de covarianza de los datos de entrada. Esta matriz muestra las relaciones estadísticas entre las diferentes variables y su variabilidad conjunta.

- **Descomposición de la matriz de covarianza:** Se realiza la descomposición de la matriz de covarianza en autovectores y autovalores. Los autovectores representan las direcciones principales o componentes principales, y los autovalores indican la importancia o la varianza explicada por cada componente principal.

- **Selección de componentes principales**: Los autovalores se ordenan en orden descendente, lo que indica la importancia relativa de cada componente principal. Luego, se selecciona un número de componentes principales para retener, generalmente basado en la varianza acumulada explicada o algún umbral predefinido.

- **Proyección de los datos en el nuevo espacio:** Los datos se proyectan en el espacio definido por los componentes principales seleccionados. Esto se logra multiplicando la matriz de datos de entrada por la matriz de autovectores correspondientes a los componentes principales seleccionados.

Esta información es tomada de [18].

6. METODOLOGÍA

6.1. Entendimiento del Problema

Una entidad financiera colombiana cuenta con un servicio de banca móvil que denominaremos **Serv-Tx-Cob**, el cual le proporciona los usuarios una forma sencilla y segura de realizar transacciones financieras utilizando su teléfono móvil. Algunos de los servicios que ofrece son:

- Transferencias de dinero: Los usuarios pueden enviar y recibir dinero de manera rápida y sencilla a través de la plataforma. Esto incluye transferencias entre cuentas Serv-Tx-Cob, transferencias a cuentas de otros bancos en Colombia e incluso transferencias internacionales.
- Pagos de servicios: Serv-Tx-Cob permite realizar pagos de servicios básicos como el agua, la luz, el gas y el teléfono de forma electrónica. Los usuarios pueden vincular sus facturas a su cuenta Serv-Tx-Cob y realizar los pagos de manera ágil y segura desde la aplicación móvil.
- Recarga de celulares: Con Serv-Tx-Cob, los usuarios pueden recargar su saldo de telefonía móvil de forma rápida y cómoda. Pueden recargar su propio número o el de familiares y amigos directamente desde la aplicación, sin necesidad de acudir a un punto de recarga físico.
- Retiros de efectivo: Serv-Tx-Cob permite realizar retiros de efectivo en cajeros automáticos o en puntos de retiro autorizados. Los usuarios pueden acceder a su dinero de manera conveniente sin necesidad de acudir a una sucursal bancaria.
- Consulta de saldo y movimientos: Los usuarios de Serv-Tx-Cob pueden consultar el saldo disponible en su cuenta y revisar los movimientos de su cuenta en tiempo real. Esto brinda un mayor control y seguimiento de las transacciones realizadas.
- Pagos en establecimientos: Serv-Tx-Cob ofrece la posibilidad de realizar pagos en comercios afiliados a través de códigos QR o tecnología NFC (Tecnología de comunicación electrónica). Los usuarios pueden realizar compras de manera segura y sin la necesidad de utilizar efectivo o tarjetas físicas.

A través de Serv-Tx-Cob se pueden realizar una gran variedad de transacciones por clientes naturales como jurídicos, tanto nacionales como extranjeros. Ahora bien, alrededor del 90 % de los usuarios de está herramienta son clientes naturales de nacionalidad colombiana, por lo cual, nos centraremos en este tipo de clientes.

Para considerar una anomalía transaccional se debe establecer un umbral, este puede variar según diferentes factores, como el tipo de transacción, el contexto empresarial y las políticas de detección de anomalías de cada entidad, no existe un monto específico establecido universalmente que determine automáticamente si una transacción es considerada una anomalía.

Establecer este umbral es un proceso que requiere un análisis cuidadoso y consideración de los factores previamente mencionados. Cada entidad puede tener sus propias políticas y enfoques para definir qué se considera una anomalía transaccional en función de su contexto y necesidades específicas. En este ejercicio se establece que el umbral es de $\$2'000{,}000$ de pesos colombianos, monto ligeramente inferior al valor de dos salarios mínimos legales mensuales vigentes, se selecciona este valor basado en la distribución del monto transaccional en el periodo de estudio para posteriormente ser validado con los expertos de negocio. Al momento de agregar las transacciones entre parejas de individuos se calcula el monto total de las transacciones en el periodo de tiempo y se compara con el umbral definido, en caso de ser menor, se excluye de este trabajo.

Los datos utilizados en este trabajo son tomados de los sistema de información de la entidad financiera, por lo cual son de carácter confidencial.

6.2. Adquisición de Datos

El conjunto de datos inicial corresponde al detalle de las transacciones efectuadas durante tres meses del año 2022 a través del servicio de banca móvil de la entidad financiera.

Es importante aclarar que estos datos deben ser tratados adecuadamente para identificar las partes y contrapartes de cada transacciones. Como no se cuenta con una herramienta adecuada para explorar y/o analizar todo el espectro transaccional, se realizan una serie de filtros sobre los datos para limitar el volumen de información.

En el periodo de tiempo seleccionado se realizaron un total de 48'899.870 transacciones, de las cuales se identificaron parcialmente 35'848.914 y se identificaron plenamente 29'982.828, es decir, se identifico la parte y contraparte de la transacción. Ahora bien, tras realizar los filtros correspondientes sobre los datos y agregar las transacciones por parejas de individuos se obtiene un total de 71'933 datos, los cuales se traducen en 101'951 individuos, por lo cual tenemos dos conjuntos de datos que se complementan entre si.

Por un lado, tenemos el conjunto de datos A, que contiene la parte y contraparte de las transacciones junto con su monto agregado, el cuál se usará para construir el grafo transaccional y el conjunto de datos B, que contiene atributos del cliente y de la transacción, el cuál se usará al momento de entrenar los modelos de detección de anomalías.

6.3. Análisis Exploratorio de Datos

El conjunto de datos A contiene 71'933 registros para las siguientes variables:

Variable	Descripción
nro telefono 1	Número de teléfono del cliente que transfiere dinero
nro identificacion 1	Número de identificación del cliente que transfiere dinero
tipo identificacion 1	Tipo de identificación del cliente que transfiere dinero
nro telefono 2	Número de teléfono del cliente que recibe dinero
nro identificacion 2	Número de identificación del cliente que recibe dinero
tipo identificacion 2	Tipo de identificación del cliente que recibe dinero
valor total tx	Monto total de las transacciones del cliente 1 al cliente 2

Cuadro 1: Descripcion Conjunto de Datos A

Tenga en cuenta que si el cliente 2 también le envía dinero al cliente 1, entonces habrá otro registro en el conjunto de datos en donde cambiaran de papeles. A partir de las 3 variables asociadas a cada cliente se genera una llave sintética que permitirá realizar los diferentes cruces de información con el conjunto de datos B, el cual contiene 101'951 registros para las siguiente variables:

Variable	Descripción	Tipo
Cod Individuo	Llave sintética	Identificador
Flg Pasivo	Indica si el cliente tiene o no productos del pasivo	Categórica Nominal
Flg Activo	Indica si el cliente tiene o no productos del activo	Categórica Nominal
Flg Listas	Indica si el cliente está o no en las listas negras	Categórica Nominal
Flg Excluidos	Indica si el cliente es o no excluido	Categórica Nominal
Profesión	Profesión del cliente	Categórica Nominal
Edad	Edad del cliente	Cuantitativa discreta
Genero	Genero del cliente	Categórica Nominal
Seg 1	Segmentación Rentabilidad	Categórica Ordinal
Seg 2	Segmentación Comercial	Categórica Nominal
Seg 3	Segmentación Relacional	Categórica Ordinal
Seg 4	Segmentación Tipo Cliente	Categórica Nominal
Seg 5	Segmentacion Etaria	Categórica Ordinal
Score Integral	Puntaje de prioridad del cliente	Cuantitativa Continua
Ingresos	Ingresos del cliente	Cuantitativa Continua
Monto Tx Recibidas	Monto total de transacciones recibidas	Cuantitativa Continua
Cant Tx Recibidas	Cantidad total de transacciones recibidas	Cuantitativa Continua
Monto Tx Enviadas	Monto total de transacciones enviadas	Cuantitativa Continua
Cant Tx Enviadas	Cantidad total de transacciones enviadas	Cuantitativa Continua
Saldo Promedio	Saldo promedio	Cuantitativa Continua

Cuadro 2: Descripción Conjunto de Datos B

Es clave tener presente el tipo de variable y la escala de medición de cada una de las variables, ya que con base en esto se implementaran ciertas técnicas de preprocesamiento antes de aplicar técnicas de Machine Learning. Este conjunto de datos está compuesto por variables sociodemográficas (profesión, edad, genero), variables de pertenencia a grupos (flg Pasivo, flg Activo, flg Listas y flg Excluidos), variables de negocio (seg1, seg2, seg3, seg4, seg5 y score integral) que han sido definidas por la entidad financiera previamente, y, variables financieras (ingresos, montos de transacciones y saldo promedio). También basado en la tipología de las variables se realizará un análisis descriptivo numérico y gráfico que nos permitirá entender su comportamiento para posteriormente caracterizar a clientes que posiblemente estén involucrados en transacciones anómalas.

6.4. Herramientas

La entidad financiera cuenta con los servicios de Google Cloud (Nube de Google), que es una plataforma que ha reunido todas las aplicaciones de desarrollo web que Google estaba ofreciendo por separado. Es utilizada para crear ciertos tipos de soluciones a través de la tecnología almacenada en la nube y permite por ejemplo destacar la rapidez y la escalabilidad de su infraestructura en las aplicaciones del buscador. Google ofrece una variedad de servicios basados en la nube, entre ellos, Google BigQuery, que es un almacén de datos para empresas que resuelve este problema, ya que permite realizar consultas de SQL de alta velocidad mediante el poder de procesamiento de la infraestructura de Google. [19]

Además cuenta con los servicios de Dataiku DSS. Esta es una herramienta de Data Science, cuya función principal es la de poder ayudar a los diferentes roles de la empresa a trabajar, modelar y presentar todo tipo de datos ya sean técnicos, analíticos o de negocio. Todo esto gracias a su uso colaborativo, donde cualquiera de los roles puede participar en las diferentes partes del proceso.

Se trata de una herramienta visual donde es posible trabajar, mediante workflows, con grandes cantidades de datos obtenidos desde multitud de fuentes. Podemos subir nuestros propios archivos de datos en formato csv, conectar a bases de datos SQL o trabajar con un gran número de conectores externos donde destacan los de Google Cloud Storage o Amazon S3 entre otros.

Una vez obtenidos los datos, la herramienta permite explorar, preparar, enriquecer, mezclar o limpiar datos de manera sencilla gracias a su interfaz. Dataiku ofrece una gran cantidad de gráficas y

opciones a la hora de visualizar datos, ya sea para uso propio o bien para la presentación de informes. Uno de los puntos fuertes es el de modelado de datos, gracias a esto se pueden definir modelos ya predefinidos de Machine Learning o bien programar algoritmos, para ello disponemos del lenguaje Python o R. [20]

6.5. Grafo Transaccional

La construcción de un grafo transaccional en el que los nodos representan individuos y las aristas están ponderadas por el monto total de las transacciones en un periodo de tiempo específico implica varias etapas:

1. Identificación de los clientes: Identifica a cada individuo único que participa en las transacciones y crea un nodo correspondiente para cada uno de ellos en el grafo.

2. Creación de enlaces ponderados: Crea enlaces entre los nodos de los individuos involucrados en cada transacción. El peso de cada enlace se basa en el monto total de las transacciones realizadas entre los individuos correspondientes en el período de tiempo dado. Es decir, el peso del enlace será el monto total acumulado de todas las transacciones entre esos individuos.

3. Representación del grafo: Utiliza una estructura de datos adecuada para representar el grafo transaccional con nodos individuales y enlaces ponderados. Esto se hace utilizando la bibliotecas de grafos NetworkX en Python, la cual proporciona funcionalidades para crear y manipular grafos.

4. Caracterización del grafo: Se describe el grafo a partir del cálculo y el análisis de las métricas de integración, segregación, centralidad y resiliencia. Además, se presentan las características básicas del grafo, el número de nodos, el número de aristas y la distribución de grados.

6.6. Algoritmo Node2Vec

El algoritmo Node2vec acepta como parámetros el número de caminatas aleatorias (**num-walks**), el tamaño de la caminatas aleatorias (**walk-length**), p y q que corresponden a los parámetros del algoritmo que genera las caminatas aleatorias, y, el número de dimensiones (dimensiones del espacio embedding).

Para optimizar estos parámetros se usará la librería Optuna y se selecciona la **Similitud de coseno** como métrica de evaluación, la cual mide la similitud de coseno entre los vectores nodos aprendidos y puede utilizarse para evaluar qué tan bien los nodos similares se agrupan juntos en el espacio vectorial. Con eso en mente se implementa la siguiente estrategia:

1. Importar las librerías necesarias, como Optuna y las funciones requeridas para calcular la similitud de coseno.

2. Definir una función objetivo que toma como argumento un objeto *trial* de Optuna. Dentro de está función, se definen posibles valores para los parámetros que queremos optimizar.

3. Entrenar el modelo Node2Vec con los parámetros definidos. Luego, se obtienen los embeddings de los nodos del modelo.

4. Calcular la matriz de similitud de coseno entre los embeddings utilizando las funciones proporcionadas. Esto nos dará una medida de similitud entre cada par de nodos.

5. Calcular la media de la similitud de coseno a partir de la matriz para obtener una medida resumen de similitud.

6. Devolver la medida de similitud de coseno como la métrica objetivo que se busca para maximizar en la optimización.

7. Crear un objeto de estudio de Optuna y especificar la dirección de maxímización para la optimización.

8. Llamar el método *optimize* del objeto de estudio de Optuna y pasar la función objetivo y el número de iteraciones que se deben realizar.

9. Obtener los mejores parámetros encontrados y entrenar la útlima versión del modelo Node2Vec. Finalmente, se obtienen los embeddings de los nodos de este modelo.

La similitud de coseno es una métrica ideal para optimizar los hiperparámetros del algoritmo node2vec en el contexto de redes o grafos debido a su enfoque en la orientación estructural relativa de los vectores, su invarianza ante la magnitud y su eficiencia computacional. Además de ello, es particularmente útil cuando se trata de datos donde la magnitud de las características es importante, pero la dirección en el espacio de características es más relevante.

6.7. Feature Engineering

En etapas anteriores hemos identificado las variables características de los nodos como lo son algunos atributos numéricos y categóricos, se calcularon algunas métricas de centralidad del grafo, y, se implemento el algoritmo Node2vec que nos permite obtener los embeddings, los cuales representan características latentes aprendidas para cada nodo. Las actividades descritas anteriormente nos han permitido construir un conjunto de datos que será insumo para la implementación de modelos de Machine Learning. Sin embargo, antes de ello, se deben implementar algunas actividades de preprocesamiento sobre los datos:

- **One-Hot Encoding**: Está técnica se utiliza para convertir las variables categóricas en una representación numérica. Cada valor único en la variable se convierte en una nueva columna binaria. Si un registro tiene un valor específico en esa columna, se establece en 1, de lo contrario, se establece en 0.

- **Ordinal Encoding**: Está técnica también se utiliza para convertir variables categóricas en una representación numérica, pero en este caso se asignan valores numéricos en función del orden o jerarquía de las categorías. A cada categoría única se asigna un valor entero único en función de su posición ordinal.

- **Robust Scaler**: Es una técnica de escalado que se utiliza para normalizar características numéricas en un rango específico, teniendo en cuenta la presencia de valores atípicos o extremos en los datos. Utiliza la mediana y el rango intercuartílico para escalar los datos.

Con base en la naturaleza de las variables, descrita en secciones anetiores, se aplica una de estas técnicas según corresponda. Estas técnicas son importantes para garantizar que los datos sean procesados de manera adecuada y coherente, evitando suposiciones erróneas y optimizando el rendimiento de los modelos de Machine Learning.

6.8. Detección de anomalías

Los modelos de detección de anomalías Isolation Forest, HBOS y Angle Base Outlier Detection se implementaron de acuerdo a la siguiente estrategia:

1. Crear la instancia del modelo con la función, y, definir una grilla con los hiperparámetros y los rangos de búsqueda.

2. Definir la métrica de evaluación: Puntaje promedio de anomalía. Realizar la búsqueda aleatoria y la validación cruzada con la función $RandomizedSearchCV()$.

3. Entrenar el modelo con los mejores parámetros encontrados.

4. Obtener y normalizar los puntajes de anomalías.

La función RandomizedSearchCV es una técnica de búsqueda aleatoria de hiperparámetros que se utiliza comúnmente en Machine Learning para optimizar los modelos. Permite buscar en un espacio de hiperparámetros definido de forma aleatoria y seleccionar la mejor combinación de ellos mediante

la validación cruzada.

Cuando se trata de modelos de detección de anomalías, es esencial utilizar una métrica adecuada para evaluar el rendimiento del modelo. Frecuentemente, la métrica de puntuación promedio de anomalía se utiliza en estos casos. La razón principal es que la detección de anomalías se basa en identificar instancias inusuales o raras en comparación con el comportamiento normal del conjunto de datos.

La métrica de puntuación promedio de anomalía asigna un puntaje a cada instancia en función de su nivel de anomalía. Esta métrica proporciona una medida cuantitativa del grado de anomalía de cada instancia y permite establecer un umbral para clasificarlos como anómalos o normales.

Al usar RandomizedSearchCV en combinación con la métrica de puntuación promedio de anomalía, se pueden explorar diferentes combinaciones de hiperparámetros del modelo de detección de anomalías y encontrar la configuración óptima que optimice el rendimiento del modelo en la detección de anomalías. La búsqueda aleatoria de hiperparámetros permite explorar un amplio espacio de búsqueda, lo que aumenta las posibilidades de encontrar una configuración que se ajuste bien a los datos y mejore la capacidad del modelo para identificar anomalías.

Los modelos implementados son algoritmos populares utilizados para la detección de anomalías. Si comparamos cómo interpretan la puntuación de anomalía en cada uno de estos modelos, encontramos las siguientes características:

- Isolation Forest: La puntuación de anomalía se calcula utilizando el número de divisiones necesarias para aislar una instancia particular. En la implementación del modelo en Python, los valores de puntuación de anomalía se normalizan y se escalan en el rango $[-1, 1]$ para facilitar la interpretación, donde un valor cercano a -1 indica una alta probabilidad de ser una anomalía y un valor cercano a 1 indica una alta probabilidad de ser una instancia normal dentro del conjunto de datos.
- HBOS: La puntuación de anomalía se calcula como una medida inversa de la densidad estimada en el bin correspondiente a una instancia dada. Cuanto menor sea la densidad estimada en el bin, más anómala se considerará la instancia. La puntuación de anomalía se normaliza y se asigna en el rango [0, 1], donde un valor cercano a 0 indica una alta probabilidad de ser una anomalía y un valor cercano a 1 indica una alta probabilidad de ser una instancia normal dentro del conjunto de datos.
- ABOD: La puntuación de anomalía se calcula a partir de la estimación de los ángulos de los vecinos, luego se combinan utilizando un enfoque de promedio ponderado. Esta puntuación no tiene un rango definido, ya que depende de los datos con los que se esté trabajando. Generalmente, se considera que los puntos con score de anomalía más altos son más propensos a ser anómalos, ya que indican una mayor desviación de la estructura de los datos.

Como el objetivo de este trabajo es construir un score de anomalía robusto es importante conocer las escalas de los scores de cada uno de los modelos de detección de anomalías. De cualquier forma es responsabilidad del investigador interpretar y definir un umbral adecuado para clasificar las instancias como anómalas o normales según los valores de puntuación de anomalía.

Teniendo en cuenta que los modelos implementados son no supervisados, hay algunas estrategias comunes para definir este umbral, entre ellas, definir el umbral manualmente según el conocimiento del investigador acerca de los datos, o, utilizar la distribución de puntuaciones de anomalía para determinar un umbral basado en cuantiles.

6.9. PCA Anomaly Scores

Para implementar el algoritmo PCA sobre los scores de anomalía de los tres modelos de detección de anomalías se adopta la siguiente estrategia:

1. Construir un dataframe en donde se cuente con el identificador del individuo y los scores de los tres modelos entrenados.
2. Visualizar las relaciones entre las variables.
3. Implementar algoritmo PCA y conservar solo la primera componente.
4. Almacenar los valores de la primera componente como una nueva variable y normalizarlos para facilitar su interpretación.
5. Generar una variable sintética en función de los cuantiles de la variable $PCA - 1$.
6. Definir una lista de posibles umbrales basada en los cuantiles e identificar la cantidad de anomalías a medida que varía el umbral escogido.
7. Seleccionar un umbral basado en un criterio de negocio.
8. Clasificar los individuos como instancias anómalas o normales.

Una vez se ha clasificado a los individuos como instancias anómalas se procede a realizar un análisis en detalle de los atributos de estos individuos.

7. RESULTADOS

7.1. Grafo Transaccional

Se implementó la estrategia planteada en la sección 6.5. A continuación, se presentan los resultados.

El grafo transaccional de personas naturales de nacionalidad colombiana está compuesto por 101'951 nodos y 71'933 aristas ponderadas por el monto total de transacciones realizadas en un periodo de tres meses.

Como las transacciones han sido agregadas por parejas de nodos, el grafo se caracteriza por ser dirigido y ponderado. Para poder representar gráficamente este tipo grafo se hace uso del objeto *DiGraph*() de la libería NetworkX, al cual se le pueden añadir el conjunto de nodos, el conjunto de aristas y los pesos de las aristas.

A continuación se presenta una ilustración del grafo:

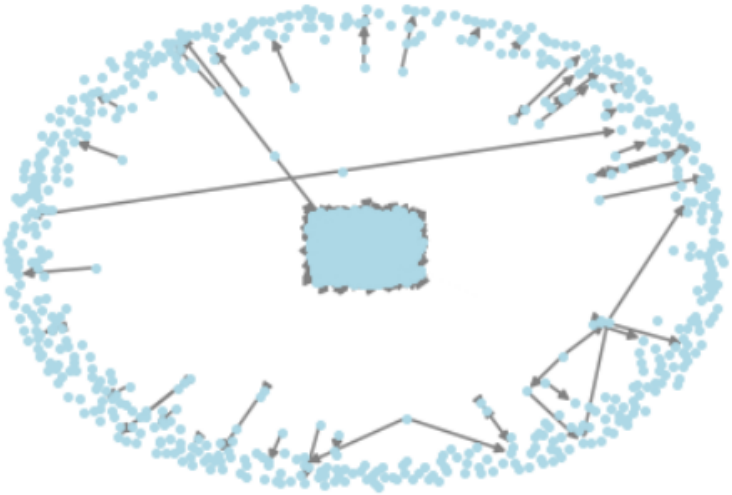

Figura 7: Ilustración del grafo

En el grafo se observan tan solo 4959 del total de nodos del grafo, llama la atención la interacción de los nodos en el centro de la red, este fenómeno podría explorarse en detalle.
A continuación se presentan algunas métricas de resumen del grafo:

Métrica	Valor
Maximum Degree	10
Minimum Degree	1
Average Degree	1.41
Median Degree	1

Cuadro 3: Métricas de integración

El grado de un nodo se refiere al número de aristas o conexiones que tiene ese nodo en un grafo. En nuestro caso, el valor máximo del grado es de 10, es decir, un individuo interactúa con otros 10 individuos como máximo. El valor mínimo del grado es de 1, es decir, existen individuos que interactúan solo con otro individuo. En promedio los individuos solo interactúan con 1 individuo. Tan solo el 28 %

de los individuo interactúan con más de un individuo.

A continuación se presentan algunas métricas de conectividad del grafo:

Métrica	Valor
Strongly Connected Components	98883
Weakly Connected Components	52105

Cuadro 4: Métricas de conectividad

La métrica del número de componentes fuertemente conectadas en un grafo se refiere al número de subgrafos fuertemente conectados que existen dentro del grafo, una componente fuertemente conectada es un subconjunto de nodos en el que cada par de nodos está conectado por un camino dirigido. Una componente fuertemente conectada se define como un grupo de nodos donde existe un camino dirigido desde cada nodo del grupo hacia cualquier otro nodo del grupo. Esto implica que todos los nodos de una componente fuertemente conectada están mutuamente alcanzables mediante aristas dirigidas. Esta métrica proporciona información sobre la estructura y conectividad del grafo, un valor alto de esta medida indica que el grafo está compuesto por múltiples subgrafos fuertemente conectados, lo que sugiere la existencia de diferentes grupos de nodos que están densamente interconectados entre sí, pero no necesariamente con los nodos de otros grupos.

Por otra parte, la métrica del número de componentes débilmente conectadas en un grafo se refiere al número de subgrafos débilmente conectados que existen dentro del grafo. Una componente débilmente conectada es un subconjunto de nodos en el que cada par de nodos está conectado por un camino, independientemente de si el camino es dirigido o no dirigido. En un grafo dirigido, una componente débilmente conectada se define como un grupo de nodos donde existe un camino que puede ser dirigido o no dirigido entre cada par de nodos del grupo. Esto implica que todos los nodos de una componente débilmente conectada están alcanzables entre sí, ya sea a través de aristas dirigidas o aristas no dirigidas. Esta métrica proporciona información sobre la estructura de conectividad general del grafo. Un valor alto de esta medida indica que el grafo está compuesto por múltiples subgrafos débilmente conectados, lo que sugiere la existencia de diferentes grupos de nodos que pueden estar conectados de forma más flexible y no necesariamente a través de caminos dirigidos.

A continuación se presentan algunas métricas de segregación del grafo:

Métrica	Valor
Transitivity	0.01556
Average Clustering Coefficient	0.00135

Cuadro 5: Métricas de segregación

La métrica de transitividad en un grafo se refiere a la proporción de triángulos cerrados en relación con el total de triángulos posibles en el grafo. En nuestro caso, la transitividad es tan solo del 0.01556, lo cual sugiere que no existe una tendencia clara de los nodos en formar agrupaciones o comunidades.

Por otra parte, la métrica del coeficiente de agrupamiento promedio en un grafo se refiere a la medida promedio de la tendencia de los nodos en el grafo a formar agrupaciones o comunidades locales. Indica cuántos de los vecinos de un nodo están conectados entre sí en comparación con la cantidad total de conexiones posibles entre los vecinos. En nuestro caso, la transitividad es tan solo del 0.00135, lo cual sugiere que no existe una tendencia clara de los nodos en formar agrupaciones o comunidades.

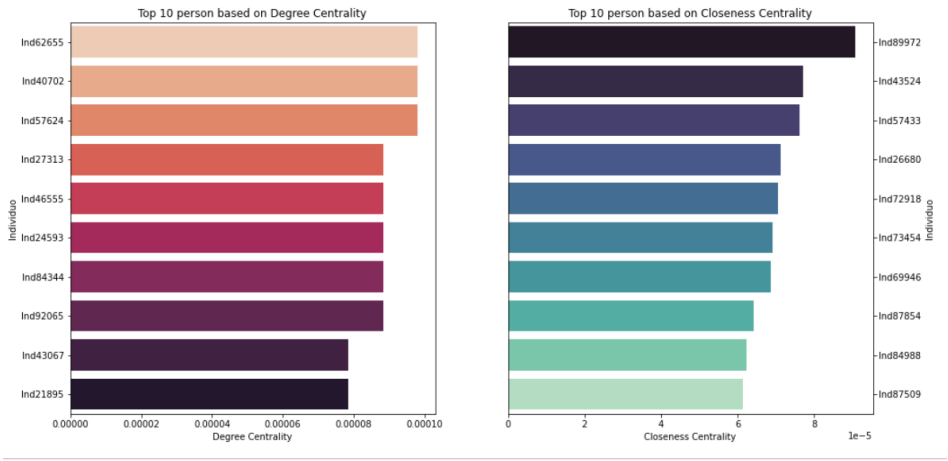

Figura 8: Métricas de Centralidad I

La métrica degree centrality (centralidad de grado) es una medida que cuantifica el número de conexiones directas que tiene un nodo en comparación con todos los demás nodos del grafo. Un nodo con un alto degree centrality tiene una gran cantidad de conexiones directas, lo que indica su importancia en términos de interacción con otros nodos en la red. En la figura 8 se presentan los 10 individuos que tienen un papel más importante en la difusión de información o influencia en la red.

Por otra parte, la métrica closeness centrality (centralidad de cercanía) es una medida que evalúa la proximidad de un nodo a otros nodos en términos de la longitud promedio de los caminos más cortos que lo conectan con todos los demás nodos. Un nodo con una alta closeness centrality se encuentra cerca de muchos otros nodos en el grafo. En la figura 8 se presentan los 10 individuos que son más eficientes en términos de comunicación y transmisión de información en la red.

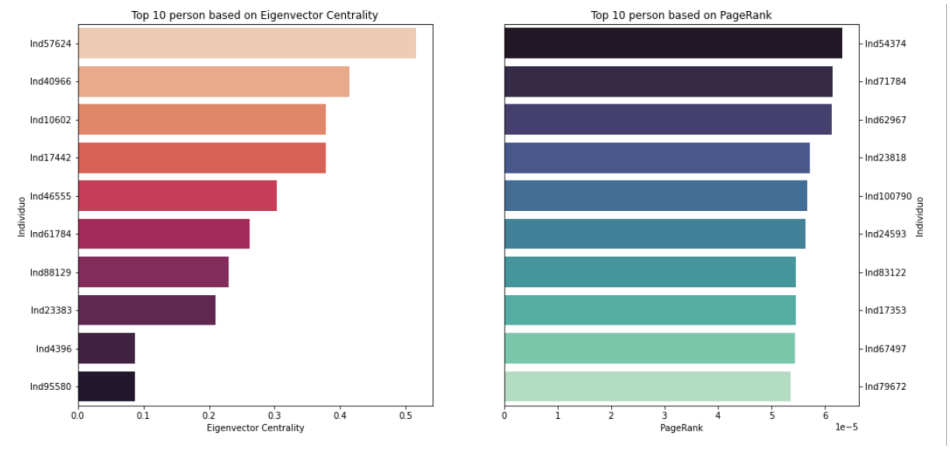

Figura 9: Métricas de Centralidad II

La métrica eigenvector centrality (centralidad de vector propio) es una medida que asigna una puntuación a cada nodo basada en la importancia de sus conexiones y la importancia de los nodos con los que está conectado. Un nodo tiene una alta eigenvector centrality si está conectado con otros nodos importantes en el grafo. Esta medida tiene en cuenta tanto la cantidad de conexiones de un nodo como la importancia de los vecinos a los que está conectado. En la figura 9 se presentan los 10 individuos que más conexiones tienen con otros nodos influyentes en la red, lo que les otorga una posición privilegiada en términos de influencia y difusión de información.

Por otra parte, la métrica PageRank asigna una puntuación a cada nodo en función de la probabilidad de que un "navegante aleatorio"llegue a ese nodo siguiendo los enlaces en el grafo. En la figura

9 se presentan los 10 individuos más importantes globalmente basados en la importancia de los nodos que se vinculan a él.

Tras construir y caracterizar el grafo, se implementa el algoritmo Node2vec con base en la estrategia planteada en la sección 6.6.

Parámetro	Posibles Valores	Mejor Valor
Num Walks	[3, 20]	12
Walk Length	[5, 100]	60
p	[0,1, 2,0]	1.1
q	[0,1, 2,0]	1.1
Nro Dimensiones	[10, 150]	128

Cuadro 6: Parámetros Node2Vec

Estos parámetros toman por defecto los siguientes valores: $numwalks = 10$, $walklength = 80$, $p = 1{,}0$ y $q = 1{,}0$, y, número de dimensiones = 128.

Para optimizar los parámetros del modelo Node2Vec se implemento una función objetivo de la librería Optuna en donde la métrica objetivo fue la similitud de coseno. Dado que Node2Vec busca optimizar los parámetros del modelo para capturar las relaciones y similitudes entre nodos en un grafo, utilizar la similitud de coseno como métrica objetivo es una elección apropiada debido a su interpretación semántica, invarianza a la magnitud y eficiencia computacional.

7.2. Detección de Anomalías

Se implementó la estrategia planteada en la sección 6.8 para los modelos Isolation Forest, HBOS y ABOD. A continuación, se presentan los resultados

Parámetro	Posibles Valores	Mejor Valor
N Estimators	{100, 200, 300}	300
Max Samples	{0,5, 0,7, 0,9}	0.9
Contamination	{0,1, 0,2, 0,3}	0.1

Cuadro 7: Parámetros Isolation Forest

Al momento de entrenar el modelo de Isolation Forest se definieron los parámetros $n\_estimators = 300$, lo que significa que se construyen 300 árboles de aislamiento, $max\_samples = 0{,}9$ que indica que se usa el 90 % de las muestras para construir cada árbol y $Contamination = 0{,}1$ que indica la proporción esperada de datos que se consideren anomalías.

Parámetro	Posibles Valores	Mejor Valor
N Bins	{10, 20, 30}	20
Alpha	{0,1, 0,5, 1,0}	0.5
Tol	{0,1, 0,5, 1,0}	0.1
Contamination	{0,1, 0,2, 0,3}	0.1

Cuadro 8: Parámetros HBOS

Al momento de entrenar el modelo HBOS se definieron los parámetros $n\_bins = 20$ que indica que se realizaran 20 divisiones en cada histograma unidimensional, $alpha = 0{,}5$ que es un factor de suavizado para evitar divisiones por cero cuando se calculan las probabilidades en los histogramas, $tol = 0{,}1$ que controla la tolerancia utilizada para determinar la convergencia del modelo, y $Contamination = 0{,}1$ que indica la proporción esperada de datos que se consideren anomalías.

Parámetro	Posibles Valores	Mejor Valor
N Neighbors	$\{10, 15, 20, 25\}$	15
Contamination	$\{0{,}1, 0{,}2, 0{,}3\}$	0.1

Cuadro 9: Parámetros ABOD

Al momento de entrenar el modelo ABOD se definieron los parámetros $method = default$ que indica que se utilizará el método preciso para calcular los ángulos entre los puntos, $n_neighbors = 155$ que indica que se consideran 15 vecinos al calcular los ángulos y $Contamination = 0{,}1$ que indica la proporción esperada de datos que se consideren anomalías.

En los tres casos la función $RandomizedSearchCV()$ se configuró para que realizará 10 combinaciones aleatorias de hiperparámetros a probar, también para que realizará 5 particiones en la validación cruzada. Además, como se menciono en la sección 6.8, se usó como métrica de evaluación la puntuación promedio de anomalía y se dijo una semilla para que los resultados sean reproducibles.

Como cada modelo entrega un score de anomalía en una escala y rango distinto, se utiliza la función $StandardScaler()$ para que las tres variables tengan media cero y una desviación estándar de uno, lo cual es útil al implementar el algoritmo PCA.

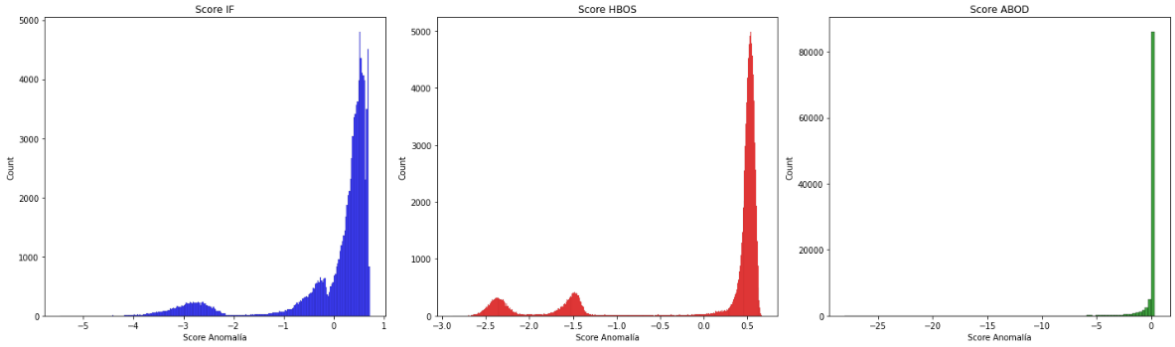

Figura 10: Scores de anomalías normalizados

7.3. PCA Anomaly Scores

Se implementó la estrategia planteada en la sección 6.9. A continuación, se presentan los resultados:

Se realizó un análisis descriptivo bivariado que permite identificar las relaciones existentes entre los scores de los modelos de detección de anomalías.

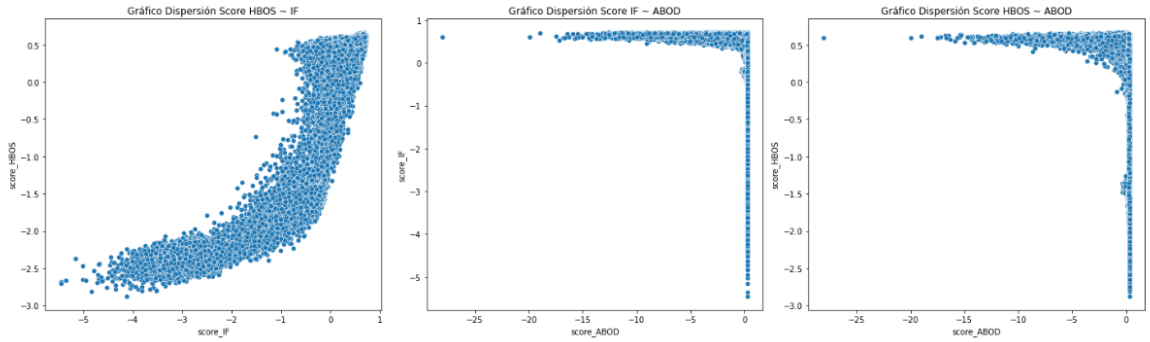

Figura 11: Diagramas de dispersión Scores

En la figura 11 se observa que no existen relaciones lineales entre las variables. Los scores de anomalías de los modelos HBOS e Isolation Forest parecen tener una relación exponencial, los scores

de anomalías de los modelos Isolation Forest y ABOD parecen tener una relación logarítmica negativa, al igual que los scores de anomalías de los modelos HBOS y ABOD.

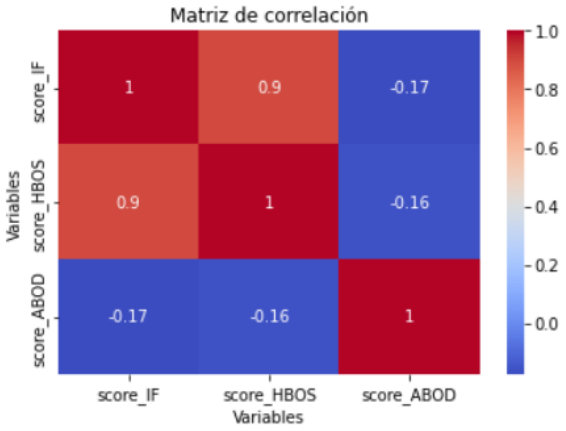

Figura 12: Matriz de correlación

Ahora bien, al calcular el coeficiente de correlación de Pearson, los scores de anomalías de los modelos HBOS e Isolation Forest tienen un 90 % de asociación lineal positiva, los scores de anomalías de los modelos Isolation Forest y ABOD tienen tan solo 17 % de asociación lineal negativa, y, los scores de anomalías de los modelos HBOS y ABOD tienen tan solo 16 % de asociación lineal negativa.

El algoritmo PCA suele funcionar mejor cuando hay relaciones lineales entre las variables. Sin embargo, lo aplicaremos en este caso. Al retener solo la primera componente principal, se conserva el 65 % de la inercia. Posterior a ello se realiza la normalización de esta variable utilizando la función $MinMaxScaler()$ que nos devuelve valores entre 0 y 1000, y, que facilita la interpretación del score robusto de anomalía.

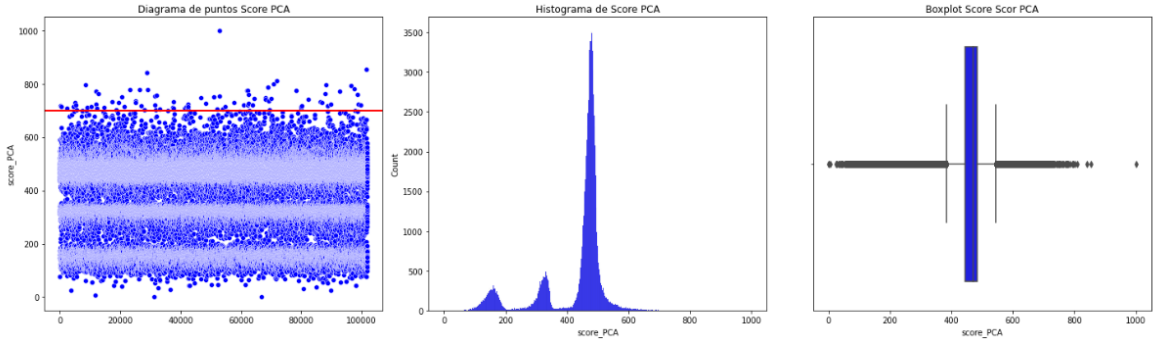

Figura 13: Análisis Descriptivo Score PCA

Si el score PCA, como hemos denominado al score robusto, toma valores muy cercanos a 0 nos alerta de la existencia de datos atípicos en la cola inferior de la distribución, mientras que si toma valores cercanos a 1000, nos alerta de la existencia de datos atípicos en la cola superior de la distribución. Como el enfoque principal de este trabajo es identificar posibles transacciones anómalas basadas en el monto total de las transacciones en un periodo específico de tiempo, nos centraremos en el análisis de los valores atípicos superiores.

En la figura 13 se observa que la distribución de la variable tiene valores atípicos en ambas colas. Se observa que la distribución de los datos podría tender a aproximarse con una distribución normal, sin embargo, resalta que hay acumulaciones de datos en dos zonas de la cola inferior de la distribución.

Ahora, se realiza un análisis de los deciles de las puntuaciones de anomalía, este análisis nos permitirá definir un umbral apropiado para considerar un dato como anómalo o no.

Percentil	10	20	30	40	50	60	70	80	90
Score	277	344	455	465	472	477	482	488	497
Nro. Anomalías	91751	81575	71094	61017	50283	40432	30220	19752	10279

Cuadro 10: Análisis de deciles

En el cuadro 10 se presenta los posibles valores para el umbral basado en los deciles de la distribución de los scores PCA. Se observa que si la elección se basa en el percentil 90, es decir, contemplando una tasa de contaminación del 10 %, el umbral sería de 497. Luego, cualquier individuo con score superior a este valor sería catalogado como una anomalía. Sin embargo, al analizar la figura 13 se observa que el bigote superior del diagrama de caja toma un valor aproximado de 544, lo cual sugiere que el umbral debe ser más alto.

Percentil	90	91	92	93	94	95	96	97	98	99
Score	497	499	502	505	509	515	522	533	550	580
Nro. Anomalías	10279	9263	8084	7159	6189	5086	4126	3067	2046	1025

Cuadro 11: Análisis de percentiles

En el cuadro 10 se presenta los posibles valores para el umbral basado en los percentiles superiores al 90 % de la distribución de los scores PCA. Si la decisión se toma basado en la medida del rango intercuartílico, entonces el umbral correspondería al percentil 98, cuyo valor es de 550. Sin embargo, para tener ambos elementos en cuenta a la hora de tomar la decisión, se opta por fijar el umbral en el percentil 97, el cual toma un valor de 533 y de esa manera clasifica a 3067 individuos como anómalos, es decir, el 3 % de los nodos del grafo.

7.4. Caracterización

Ahora, una vez clasificado a 3067 individuos como anómalos se debe hacer una revisión exhaustiva de las variables presentadas en la sección 6.3 con el objetivo de caracterizar a cada individuo y poder determinar si realmente debe considerarse que esté involucrado en transacciones anómalas.

A continuación se presentan algunas generalidades encontradas:

- Tan solo el 8 % de los 3067 individuos (231) tienen una cuenta de ahorros y/o corriente activa con la entidad financiera y ninguno de ellos tiene productos crediticios activos.
- El 61 % de los individuos son mujeres, mientras que el 39 % restante son hombres.
- Todos los individuos se caracterizan por ser no rentables. Es decir, individuos que generan menos beneficios o ingresos en comparación con los costos asociados con su adquisición, mantenimiento o servicio. En otras palabras, son individuos cuyo valor económico o contribución a la rentabilidad de una empresa es limitado o negativo.
- El 92.5 % de los individuos se caracterizan por ser individuos segmentados bajo el uso estricto del servicio de depósito electrónico, es decir, no utilizan otros servicios de la entidad financiera.
- El 64 % de los individuos son segmento A que indica que sus ingresos promedios son menores a 1 SMMLV, mientras que el 36 % restante son segmento B que indica que sus ingresos promedios son de 1 SMMLV, pero además tienen invertidos al menos 10 SMMLV en productos fiduciarios y/o Cdts.
- El 92 % de los individuos son empleados, el 5 % son personas naturales con negocios y el 3 % son independientes.
- El 63 % son individuos adultos y el 37 % restante son individuos jóvenes.

Se detecta que 792 de los individuos tienen la misma cantidad de dinero recibida ($4030739) en la misma cantidad de transacciones (12 Transacciones) durante el periodo de 3 meses, lo cual sugiere que reciben el pago de sus salarios a través del servicio de depósito electrónico. Por otra parte, se identifican 3 individuos que han recibido transacciones por un monto de hasta 9 veces sus ingresos percibidos en una cantidad importante de transacciones.

Finalmente se habilita una herramienta de código que permite describir rápidamente a cualquiera de los individuos que posiblemente estén involucrados en transacciones anómalas.

8. CONCLUSIONES Y RECOMENDACIONES

8.1. Conclusiones

En este trabajo se logró desarrollar una metodología robusta para la detección de anomalías transaccionales integrando la teoría de grafos y técnicas de machine learning no supervisadas. La metodología se describe a continuación:

Inicialmente, se realizó una exploración de las transacciones para comprender su naturaleza. Además, se identificó la parte y contraparte de cada transacción, lo que permite establecer la relación entre los individuos involucrados en las transacciones.

Luego, se agregaron los montos de las transacciones realizadas entre cada par de individuos. Esto implica sumar los montos de todas las transacciones entre dos individuos específicos en el período de tiempo de tres meses. Esta agregación de montos proporciona una medida cuantitativa del nivel de interacción financiera entre los individuos.

En paralelo, se recopilaron variables sociodemográficas y financieras relacionadas con cada individuo que realiza transacciones. Estas variables pueden incluir información como edad, género, profesión, ingresos, entre otros. Estas variables proporcionan una caracterización más completa de los individuos y pueden ser utilizadas como características adicionales en el análisis de detección de anomalías.

En otros ejercicios, la entidad financiera ha realizado clasificaciones y segmentaciones de los clientes con base en ciertos criterios. Estas clasificaciones están relacionadas con el riesgo crediticio, el comportamiento financiero, entre otros aspectos relevantes para la entidad. Estas clasificaciones proporcionan información adicional sobre el perfil y comportamiento de los clientes, lo que puede ser útil en el análisis de detección de anomalías.

Se realizó la construcción de un grafo donde los nodos representan a los individuos y las aristas representan las transacciones entre ellos. Además, se asignó un peso a cada arista basado en el monto total de las transacciones en el período de tiempo analizado. Esta representación del grafo captura la estructura de las interacciones entre los individuos y la magnitud de las transacciones. Además, se calcularon una serie de métricas de centralidad que permiten describir y caracterizar la importancia de cada nodo al interior del grafo.

Posterior a eso, se utilizó el algoritmo Node2vec para generar embeddings de los nodos del grafo. Node2vec es un algoritmo de aprendizaje no supervisado que captura las relaciones locales y globales en un grafo y produce representaciones vectoriales de los nodos. Estos embeddings capturan las características y la estructura del grafo, lo que permite realizar análisis y detección de anomalías en un espacio de menor dimensionalidad.

Luego, se implementaron tres algoritmos de detección de anomalías utilizando los embeddings generados por Node2vec. Estos algoritmos incluyeron métodos como Isolation Forest, Histogram Based Outlier (HBOS) y Angle-Based Outlier Detection (ABOD). Para cada algoritmo se calcularon sus puntuaciones de anomalía para los nodos del grafo, identificando aquellos que se desvían significativamente de los patrones normales de transacciones.

Después, se aplicó un análisis de componentes principales (PCA) utilizando las puntuaciones de anomalía obtenidas de los tres modelos anteriores. El PCA permitió reducir la dimensionalidad de los datos y capturar la variabilidad de las puntuaciones de anomalía en un espacio de menor dimensión. Esto proporcionó un score robusto de anomalía que puede utilizarse para clasificar y priorizar los nodos en función de su nivel de anomalía. Finalmente, se diseña un ejecutable que permite generar un reporte sencillo en donde se caracteriza a los individuos involucrados en transacciones anómalas, lo cual reducirá sustancialmente la carga operativa en actividades de investigación.

Como se plantea al inicio de este documento, la detección de anomalías juega un papel fundamental en la lucha contra el lavado de activos. La metodología diseñada permite detectar actividades inusuales, identificar estructuras complejas, se puede analizar grandes volúmenes de datos de manera más rápida

y precisa que los enfoques manuales tradicionales, entre otras. De esta manera, la entidad financieras puede fortalecer sus mecanismos de supervisión y control, y tomar medidas proactivas para prevenir y combatir el lavado de activos.

Este ejercicio puede ser considerado como la etapa inicial de un ejercicio robusto de detección de anomalías transaccionales. Si a raíz del ejercicio no supervisado, se comprueba la existencia de las anomalías transaccionales, se podría realizar un ejercicio supervisado, siempre basado en el conocimiento de los expertos.

8.2. Recomendaciones

A continuación se presentan una serie de recomendaciones que pueden ser útiles para futuras investigaciones:

1. Robustecer la actividad de variables sociodemograficas y financieras relacionadas con cada individuo que realiza transacciones. Se debe poder garantizar que estos datos sean precisos, completos, consistentes y confiables.

2. El análisis de grafos puede es computacionalmente costoso, especialmente cuando se trabaja con conjuntos de datos grandes y complejos. Poder realizar un análisis completo sobre el espectro transaccional de una entidad financiera requiere herramientas y algoritmos especializados.

3. Explorar alternativas al algoritmo Node2Vec, entre ellos se destacan DeepWalk, GraphSage, LINE (Large Scale Information Network Embedding), TADW (Topological Autoencoder for Directed Weighted Grapgs) y GCN (Graph Convolutional Networks). Es recomendable evaluar el rendimiento de estos algoritmos en función de los requisitos y las características específicas del conjunto de datos.

4. Explorar otro tipo de modelos de detección de anomalías no supervisados como los métodos basados en densidad, los métodos basados en distancias y los métodos basados en redes.

9. REFERENCIAS

[1] "¿qué es aml (anti-money laundering)?" urlhttps://www.tusdatos.co/blog/que-es-aml-anti-money-laundering, 2022.

[2] "Uiaf colombia: ¿qué es la unidad de información y análisis financiero?" urlhttps://www.worldsys.co/uiaf-colombia-que-es-la-unidad-de-informacion-y-analisis-financiero/, 2022.

[3] "Sarlaft: Sistema de administración del riesgo de lavado de activos y de la financiación del terrorismo," urlhttps://www.isotools.org/2018/10/08/sarlaft-sistema-administracion-riesgo-lavado-activos-financiacion-terrorismo/, 2018.

[4] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *2008 eighth ieee international conference on data mining*. IEEE, 2008, pp. 413–422.

[5] K. B. Varun and D. Bollmann, "Unsupervised anomaly detection in high-dimensional data: A survey," *Data Mining and Knowledge Discovery*, vol. 33, no. 2, pp. 337–387, 2019.

[6] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection in e-commerce transactions using machine learning techniques," in *Proceedings of the ACM SIGKDD Workshop on Data Mining for Business Applications*. ACM, 2009, pp. 64–72.

[7] J. Smith and E. Johnson, "A novel hybrid intrusion detection system based on one-class support vector machines," *Journal of Network Security*, vol. 25, no. 3, pp. 123–136, 2022.

[8] X. Du, F. Wang, and L. Wang, "Deeplog: Anomaly detection and diagnosis from system logs through deep learning," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2017, pp. 1285–1298.

[9] S. Guha, N. Mishra, S. Roy, and O. Schrijvers, "Robust random cut forest based anomaly detection on streams," *Journal of Machine Learning Research*, vol. 17, no. 22, pp. 1–40, 2016.

[10] A. Patcha and J.-M. Park, "A survey of network anomaly detection techniques," *IEEE Communications Surveys & Tutorials*, vol. 9, no. 1, pp. 48–67, 2007.

[11] R. Chalapathy and S. Chawla, "Deep learning for anomaly detection: A survey," *arXiv preprint arXiv:1901.03407*, 2019.

[12] L. Akoglu, H. Tong, and D. Koutra, "Graph based anomaly detection and description: A survey," *Data Mining and Knowledge Discovery*, vol. 29, no. 3, pp. 626–688, 2015.

[13] K. Z. Sultana, S. Jha, and S. Murtaza, "Autoencoders for unsupervised anomaly detection in network traffic data," in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2018, pp. 785–792.

[14] G. Xu, W. Chen, and J. Qian, "Network anomaly detection using recurrent neural networks," in *2018 IEEE International Conference on Big Data (Big Data)*. IEEE, 2018, pp. 3552–3557.

[15] C. Stamile, A. Marzullo, and E. Deusebio, *Graph Machine Learning*. Packt Publishing, 2021.

[16] "Algorithm selection for anomaly detection," https://medium.com/analytics-vidhya/algorithm-selection-for-anomaly-detection-ef193fd0d6d1, 2020.

[17] M. Goldstein and A. Dengel, "Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm," 2012.

[18] I. T. Jolliffe, *Principal Component Analysis*. Springer Series in Statistics, 2011.

[19] Google, "Documentación bigquery," urlhttps://cloud.google.com/bigquery/docs, 2021.

[20] Dataiku, "Documentación dataiku," urlhttps://academy.dataiku.com, 2021.