Detección de operaciones sospechosas de lavado de activos en entidades financieras, 2022

Julián Arley Chaverra

Monografía presentada para optar al título de Especialista en Analítica y Ciencia de Datos

Tutor

Daniela Serna Buitrago, Especialista (Esp)

Universidad de Antioquia

Facultad de Ingeniería

Especialización en Analítica y Ciencia de Datos

Medellín, Antioquia, Colombia

2022

Cita (Chaverra, 2022)

Referencia Chaverra, J. A. (2018). *Detección de operaciones sospechosas de lavado de activos en entidades financieras, 2022* [Trabajo de grado especialización]. Universidad de Antioquia, Medellín, Colombia.

Especialización en Analítica y Ciencia de Datos, Cohorte III

UNIVERSIDAD DE ANTIOQUIA
Vicerrectoría de Docencia
Sistema de Bibliotecas

Centro de Documentación Ingeniería (CENDOI)

Repositorio Institucional: http://bibliotecadigital.udea.edu.co

Universidad de Antioquia - www.udea.edu.co

Rector: John Jairo Arboleda Céspedes

Decano/Director: Jesús Francisco Vargas Bonilla

Jefe departamento: Diego Botia Valderrama

TABLA DE CONTENIDOS

1. RESUMEN EJECUTIVO

El lavado de activos se define como el proceso mediante el cual personas u organizaciones criminales pretenden dar apariencia de legalidad a recursos generados en actividades ilícitas, evitando que sus actividades sean detectadas por las autoridades judiciales, que no sean detectadas por los controles establecidos por las entidades que gestionan el riesgo de lavado de activos y financiación del terrorismo o ser reportados a la Unidad de Información y Análisis Financiero UIAF. En Colombia, esta práctica es un delito que se encuentra definido en el artículo 323 del Código Penal (Unidad de Inteligencia y Análisis Financiero, 2018).

Para la prevención del lavado de activos, en Colombia se han generado normas por medio de las cuales, las entidades de supervisión como la Superintendencia Financiera de Colombia, Superintendencia de Sociedades, Superintendencia de Economía Solidaria, entre otras; imparten a sus entidades vigiladas instrucciones para la administración del riesgo de lavado de activos y financiación del terrorismo LA/FT. De manera específica, las entidades financieras en Colombia deben dar cumplimiento al Capítulo IV, Título IV, Parte I de la Circular Básica Jurídica de la Superintendencia Financiera de Colombia, donde se señala la identificación de operaciones inusuales y determinación de operaciones sospechosas (Superintendencia Financiera de Colombia, 2020, #).

Esta monografía describe un modelo analítico para el análisis de la transaccionalidad en una entidad financiera y la identificación de las operaciones inusuales que deberán ser evaluadas con mayor profundidad para la determinación de las operaciones sospechosas de lavado de activos; por lo cual, el resultado esperado, es una clasificación de cada una de las transacciones señalando cuáles de ellas se consideran operaciones inusuales y cuáles no. Para obtener el modelo propuesto, primero se hace un análisis exploratorio de los datos EDA, seleccionan las métricas de desempeño con los cuales se evaluarán los modelos y por último se entrenan varios modelos para los cuales se comparan las métricas de desempeño y se selecciona aquel modelo que presente mejor rendimiento respecto a las métricas seleccionadas.

El principal obstáculo para el desarrollo de este proyecto fue la disponibilidad de los datos, pues las organizaciones no están dispuestas a compartir la información transaccional de sus clientes por el secreto bancario que aplica a esta información, motivo por el cual se acudió a una base de datos sintética. Adicionalmente, estas bases de datos presentan problemas de desbalance en los datos respecto a la variable objetivo y por ello, las métricas de evaluación seleccionadas son especiales para este tipo de casos, así como las estrategias y/o modelos propuestos.

Los notebooks del proyecto descrito en esta monografía pueden ser consultados en el siguiente repositorio de GitHub: https://github.com/julianchaver/Monografia_Deteccion_Op_Sospechosas.git

2. DESCRIPCIÓN DEL PROBLEMA

Las entidades financieras captan recursos de sus clientes y ponen a su disposición diferentes canales o plataformas para la realización de transacciones; servicios que a su vez exponen a la entidad financiera a riesgos como el fraude y el lavado de dinero, para lo cual realizan una importante inversión de recursos humanos, financieros y tecnológicos en la implementación de mecanismos para administrar y prevenir estos riesgos a los que se exponen en el ejercicio de su actividad financiera y en el mercado.

Los mecanismos para la prevención de lavado de activos y financiación del terrorismo pretenden proteger a la entidad financiera de ser utilizada como vehículo para la comisión de estos delitos por parte de los clientes y usuarios. Uno de estos mecanismos es el monitoreo o análisis de cada una de las transacciones de los clientes, identificando cuáles de ellas representan un riesgo para la entidad, es decir, determinan cuáles transacciones son inusuales y sospechosas de lavado de activos o financiamiento del terrorismo.

Si además se tiene en cuenta que, en las entidades financieras, al día se realizan cientos o miles de transacciones, que cualquier de ellas puede ser riesgosa sin importar el valor de la misma o la condición socioeconómica de quien realiza la operación; la entidad financiera debe adoptar herramientas y estrategias para realizar esta gestión con la mayor oportunidad posible, preferiblemente en tiempo real, para lograr detener la operación o tomar medidas correctivas al respecto.

2.1 PROBLEMA DE NEGOCIO

En la industria se observa con frecuencia que la identificación de operaciones inusuales y sospechosas se fundamenta en estrategias consistentes en la comparación del monto de las transacciones con la información financiera del cliente, adoptando límites a partir de los cuales se genera un alertamiento automático (reglas duras), entre otros mecanismos con los cuales obtienen tasas de falsos positivos superiores al 50% o más respecto a las transacciones alertadas. Esta situación impide que las entidades tengan una capacidad de reacción oportuna para la determinación y reporte de operaciones sospechosas o para evitar la materialización de la operación de lavado de activos.

Otra problemática que se ha identificado es la falta de herramientas y técnicas para este tipo de análisis, encontrando por ejemplo, entidades que comparan de manera manual respecto a cierto criterio predeterminado, las transacciones una a una en archivos de Excel para determinar si la operación se considera inusual o sospechosa.

2.2 APROXIMACIÓN DESDE LA ANALÍTICA DE DATOS

La problemática descrita en el numeral anterior, puede ser abordada desde la analítica de datos mediante el modelamiento del comportamiento transaccional de los clientes con un modelo de clasificación lo suficientemente robusto para procesar datos desbalanceados, que presente bajas tasas de falsos positivos y no requiera tiempos prolongados de ejecución para una oportuna detección de operaciones inusuales y sospechosas. Este tipo de modelo permite identificar patrones comportamentales o características de los clientes y transacciones que de manera directa no puedan ser percibidas por el hombre.

La aplicación de la analítica ayuda a fortalecer los mecanismos de prevención de las entidades financieras en la identificación de transacciones que representan un riesgo de lavado de activos, conocidas como operaciones inusuales y operaciones sospechosas.

Como resultado de la aplicación de estos modelos analíticos al monitoreo transaccional, se espera obtener una mayor efectividad en la identificación de las operaciones inusuales y sospechosas con bajos niveles de falsos positivos.

2.3 ORIGEN DE LOS DATOS

La información transaccional de los clientes de las entidades financieras está sujeta a la aplicación del secreto bancario por temas normativos relacionados con la seguridad de la información y la protección de datos personales; razón por la cual no fue posible obtener información transaccional real de una entidad financiera. Por lo tanto, los datos utilizados se obtuvieron de una base de datos sintética disponible en Kaggle.

La base de datos simula la dinámica transaccional de una entidad financiera, con datos recopilados de manera diaria e individual y contienen la información mínima para la identificación y trazabilidad de cada transacción como lo son: fecha, valor, cliente, beneficiario de la operación si aplica y forma como se realizó la transacción (ESEBAMEN, n.d.).

2.4 MÉTRICAS DE DESEMPEÑO

Para cumplir con el propósito de esta monografía, las métricas de desempeño el modelo de clasificación para la identificación de operaciones inusuales y sospechosas deberá garantizar:

- Alcanzar por lo menos una efectividad del 70% en la identificación de las operaciones inusuales y sospechosas reales, mejor conocido en el contexto de la analítica como la especificidad. Esta métrica indica que el modelo está lo suficientemente entrenado para diferenciar las operaciones normales de las operaciones inusuales sospechosas.
- La tasa de falsos positivos u falsas señales de alerta en la identificación de operaciones inusuales y sospechosas no deberá ser superior al 30%.

- Los resultados de las métricas con los datos de prueba y de validación deberán proporcionar la tranquilidad de que el modelo seleccionado no se encuentra sobre o sub ajustado.

3. DATOS

Es importante entender los datos con los cuales se construirán y evaluarán los modelos, saber por lo menos qué información suministra cada variable o atributo del dataset, por lo tanto, a continuación, se presenta un análisis de la base de datos para tener idea de cómo se distribuyen o comportan los datos y cómo se relacionan entre sí.

3.1 DATOS ORIGINALES

La información o datos originales utilizados para el desarrollo de este modelo tiene la particularidad de ser información confidencial y sometida a lo que se conoce como el secreto bancario (Sierra Fajardo, 2021), es decir, es información que las entidades financieras deben salvaguardar celosamente y garantizar que la misma no sea conocida por personas no autorizadas por la entidad financiera y distintas al cliente. Por esta razón, una entidad bancaria no accede a suministrar estos datos así sea de manera anonimizada, pues hasta al mismo regulador, la Unidad de Inteligencia y Análisis Financiero UIAF (https://www.uiaf.gov.co) no accede a suministrar este tipo de información ni siquiera de manera anonimizada, argumentando que la información que reposa en sus bases de datos, se encuentra protegida por la reserva legal, acorde a la Leyes 526 de 1999 y 1621 de 2013, por ser información en el marco de actividades de inteligencia y contrainteligencia.

En consecuencia, se accedió a la base de datos publicada en Kaggle (https://www.kaggle.com/x09072993/aml-detection/data) la cual suministra información de una entidad financiera con 6.362.620 registros o transacciones y 11 atributos que recopilan la información de cada transacción. Estas transacciones fueron sometidas a revisión por parte de la entidad financiera identificando cuáles fueron operaciones sospechosas y las operaciones normales respecto al riesgo de lavado de activos. A continuación, se describe cada uno de los atributos de la base de datos:

Tabla 1
Descripción de la base de datos original

Nombre original	Nombre asignado	Tipo de dato	Descripción
Step	Hora	entero	momento en que fue realizada la transacción. Según la información de la fuente de los datos, corresponde a una hora de tiempo.
Type	FormaTx	texto	forma en que fue realizada la operación, es decir, en efectivo o mediante transferencia electrónica u otra forma.
Amount	ValorTx	flotante	valor de la operación en dólares estadounidenses.
NameOrig	Cliente	texto	identificación del cliente que realiza la operación. Estas identificaciones han sido alteradas para preservar la identidad de las personas.
OldbalanceOrg	SaldoIniCliente	flotante	saldo inicial del cliente que realiza la operación.
NewbalanceOrig	SaldoFinCliente	flotante	saldo final del cliente.

NameDest	Beneficiario	texto	nombre del destinatario o beneficiario de la operación. Es hacia quién se realizó el movimiento
OldbalanceDest	SaldoIniBenef	flotante	saldo inicial del destinatario.
NewbalanceDest	SaldoFinBenef	flotante	saldo final del destinatario.
isFraud	Fraude	entero	evaluación de la transacción respecto al riesgo de lavado de dinero. Toma valores de 1 cuando es sospechosa y 0 cuando es normal. Este atributo será considerado la variable objetivo o target para el entrenamiento y validación de los modelos.
isFlaggedFraud	Fraude>200	entero	marcación de las operaciones de lavado de dinero intentadas por un monto mayor a US$200.000 Toma valores de 1 cuando la operación sospechosa es mayor a US$200.000 y 0 cuando es menor a este valor

Fuente: Información de los atributos extraídos de (ESEBAMEN, n.d.)

Figura 1

Resumen tipo de datos del DataFrame

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6362620 entries, 0 to 6362619
Data columns (total 11 columns):
 #   Column           Dtype
---  ------           -----
 0   Hora             int64
 1   FormaTx          object
 2   ValorTx          float64
 3   Cliente          object
 4   SaldoIniCliente  float64
 5   SaldoFinCliente  float64
 6   Beneficiario     object
 7   SaldoIniBenef    float64
 8   SaldoFinBenef    float64
 9   Fraude           int64
 10  Fraude>200       int64
dtypes: float64(5), int64(3), object(3)
```

Fuente: Elaboración propia.

La base de datos original puede ser consultada en la dirección electrónica de Kaggle https://www.kaggle.com/x09072993/aml-detection/data

3.2 ANÁLISIS EXPLORATORIO DE DATOS - EDA

En el notebook 01-EDA_operaciones_sospechosas.ipynb que se encuentra en el repositorio de GitHub se puede consultar en mayor detalle el análisis exploratorio y descriptivo de la base de datos. A continuación, se resaltan los aspectos más relevantes para el desarrollo de los modelos predictivos, identificando los atributos categóricos o cuantitativos para determinar más adelante el pipeline de proceso analítico.

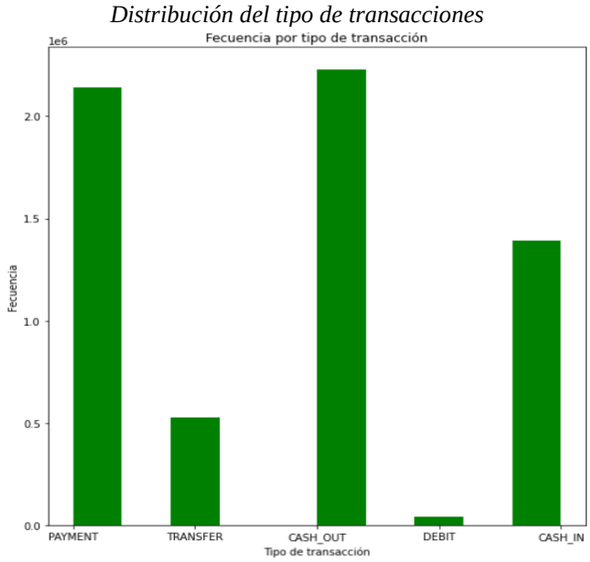
Variables categóricas

Se cuenta con dos variables categóricas de importancia para el modelo predictivo: los tipos de transacción realizados en la entidad financiera y denominada **FormaTx**, para la cual se encuentra que las formas usuales son en su mayoría salidas de dinero en efectivo, pagos y entradas en efectivo.

En menor proporción se encuentran transacciones mediante transferencia y débito, lo cual se puede apreciar a continuación:

Figura 2

Distribución del tipo de transacciones



Fuente: Elaboración propia

La otra variable categórica es la variable objetivo **Fraude**, que indica si la operación es sospechosa o normal adoptando valores de 0 cuando es una operación normal y 1 cuando la operación se considera sospechosa. Al revisar la distribución de este atributo se encuentra un desbalance importante en las etiquetas o valores de la variable objetivo en las proporciones indicadas en la siguiente tabla. Este se constituye en el mayor reto del proyecto ya que el desbalance en los datos podría afectar seriamente el desempeño de los modelos entrenados.

Tabla 2

Balanceo de clases objetivo

Valor etiqueta	Porcentaje
0	99.87%
1	0.13%

Fuente: Elaboración propia

El desbalance de datos es la presencia mayoritaria de un tipo o clase de observaciones respecto a la otra u otras clases de observaciones. El trabajar con un conjunto de datos desbalanceado sin adoptar ninguna estrategia adicional, será un problema para la métrica de medición del desempeño del algoritmo de predicción (Gonzalez, 2019). En otro apartado de este artículo se abordarán las estrategias aplicadas para el balanceo del conjunto de datos.

Variables cuantitativas

Figura 3

Resúmen estadístico variables cuantitativas

	Hora	ValorTx	SaldoIniCliente	SaldoFinCliente	SaldoIniBenef	SaldoFinBenef	Fraude	Fraude>200
count	6.362620e+06	6.362620e+06	6.362620e+06	6.362620e+06	6.362620e+06	6.362620e+06	6.362620e+06	6.362620e+06
mean	2.433972e+02	1.798619e+05	8.338831e+05	8.551137e+05	1.100702e+06	1.224996e+06	1.290820e-03	2.514687e-06
std	1.423320e+02	6.038582e+05	2.888243e+06	2.924049e+06	3.399180e+06	3.674129e+06	3.590480e-02	1.585775e-03
min	1.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
25%	1.560000e+02	1.338957e+04	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
50%	2.390000e+02	7.487194e+04	1.420800e+04	0.000000e+00	1.327057e+05	2.146614e+05	0.000000e+00	0.000000e+00
75%	3.350000e+02	2.087215e+05	1.073152e+05	1.442584e+05	9.430367e+05	1.111909e+06	0.000000e+00	0.000000e+00
max	7.430000e+02	9.244552e+07	5.958504e+07	4.958504e+07	3.560159e+08	3.561793e+08	1.000000e+00	1.000000e+00

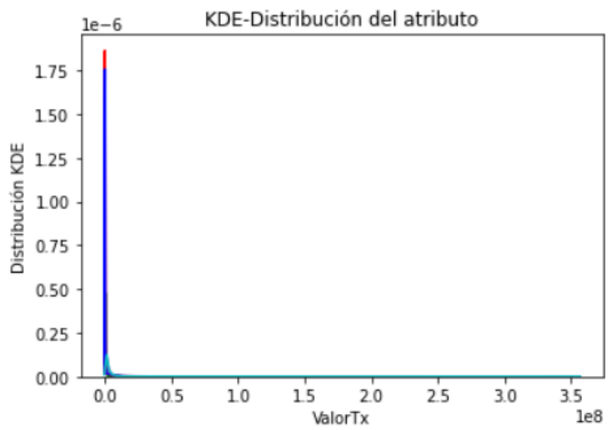
Fuente: Elaboración propia

Las otras variables utilizadas para predecir las operaciones sospechosas son: el valor de la transacción **ValorTx,** el saldo inicial y final del cliente, el saldo inicial y final del destinatario o beneficiario de la transacción. En el resumen estadístico se puede validar rápidamente que ninguna de las variables tiene valores negativos y la presencia de valores atípicos al comparar el valor máximo con su respectivo percentil 75, valores atípicos que son tratados en el preprocesamiento de los datos.

Al consultar el notebook del análisis exploratorio de datos, se puede observar que la distribución de estas cinco variables presenta una asimetría positiva o hacia la derecha (*Cómo La Asimetría Y La Curtosis Afectan La Distribución - Minitab*, n.d.), es decir, la mayoría de las observaciones presentan valores acumulados a la izquierda, pero se cuenta con otras observaciones de alto valor y en menor frecuencia que ocasionan que la curva de distribución se prolongue hacia la derecha con frecuencias muy bajas.

Figura 4

Densidad kernel de las variables cuantitativas



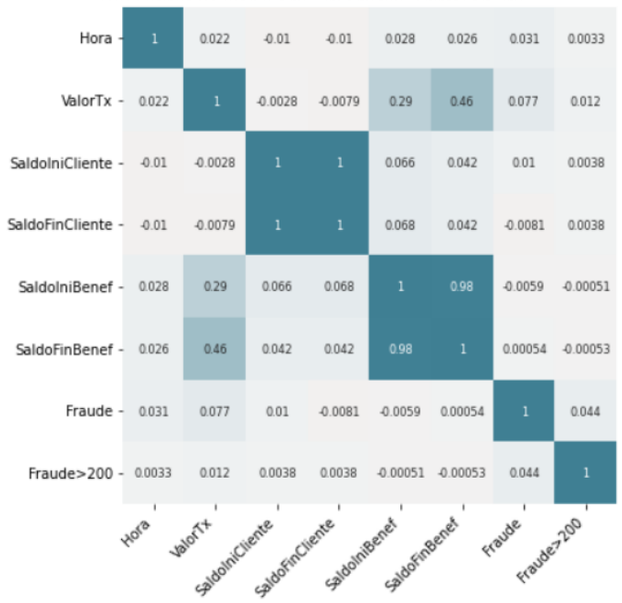
Fuente: Elaboración propia

En este momento, el mayor interés radica en seleccionar cuáles serán las variables predictoras más apropiadas para el modelo, razón por la cual se da importancia a la correlación entre estas variables con la variable objetivo. En los siguientes gráficos de correlación se aprecia que existe una fuerte correlación entre el saldo inicial y final del cliente, así como entre el saldo inicial y final del

beneficiario de la transacción, lo cual tiene sentido ya que el saldo final de un producto depende de los cambios presentados en el saldo inicial del mismo.

Figura 5

Correlación entre las variables cualitativas

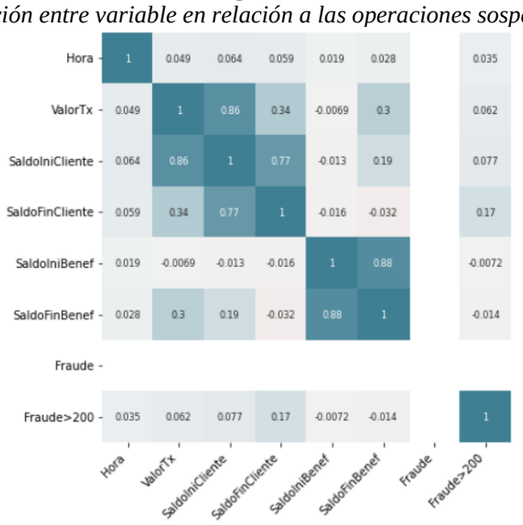


Fuente: Elaboración propia

Sin embargo, es importante hacer una precisión adicional en este análisis, ya que en la variable objetivo el mayor interés se encuentra sobre la clase 1 u operación sospechosa y, teniendo en cuenta el desbalance de los datos, en la gráfica siguiente se valida la correlación de las variables con la variable objetivo clase1, donde se puede apreciar cómo disminuye el grado de correlación entre los saldos iniciales y finales tanto del cliente como del beneficiario, y el valor de la transacción (principal variable en concepto del autor), aumenta de manera significativa con el saldo inicial del cliente, resultado esperado por el autor ya que el valor de la transacción depende del saldo que tiene disponible el cliente para realizar una transacción.

Figura 6

Correlación entre variable en relación a las operaciones sospechosas



Fuente: Elaboración propia

Se podría continuar profundizando en el análisis descriptivo univariado y multivariado de cada uno de los atributos del dataset, pero para el objetivo de este proyecto, la descripción anterior permite

conocer la información suficiente y relevante para saber cómo se comportan y relacionan los atributos entre sí y cuáles pueden ser más relevantes para el entrenamiento del modelo de clasificación para predecir las operaciones sospechosas.

3.3 DATASETS

Para el entrenamiento y validación de los modelos que se describen en este artículo, el set de datos original, luego de haber aplicado las transformaciones requeridas y descritas más adelante, se subdividió en cuatro conjuntos de datos de manera aleatoria y estratificada, o sea, manteniendo la misma proporción de las clases en la variable objetivo (*3.1. Cross-Validation: Evaluating Estimator Performance — Scikit-Learn 1.0.2 Documentation*, n.d.): un conjunto de entrenamiento con las variables predictoras, un conjunto de entrenamiento de la variable objetivo, un conjunto de validación con las variables predictoras y el conjunto de validación con la variable objetivo. Los conjuntos de entrenamiento contienen el 70% de los datos de la variable original y los conjuntos de validación contienen el 30%.

Posteriormente, para compensar o reducir el impacto que se tendría en la efectividad de los modelos, se aplica una estrategia de balanceo de datos mediante el resampleo de los conjuntos de entrenamiento en las variables predictoras y en la variable objetivo. La estrategia aplicada se denomina subsampling o submuestreo de la clase mayoritaria, la cual consiste en reducir la clase mayoritaria basándose en un algoritmo similar al k-nearest neighbor para seleccionar las muestras o registros a eliminar (Na8, 2019). En la cita referenciada se explican otras estrategias, pero se seleccionó esta por las siguientes razones:

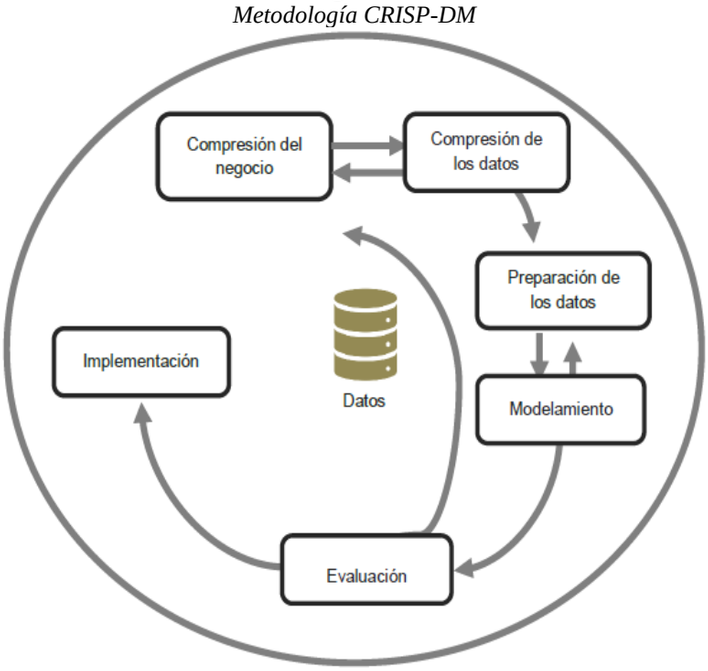
- Se conservan los registros originales de la clase objetivo minoritaria.
- El submuestreo permite reducir el tamaño de los conjuntos de datos de entrenamiento en la clase mayoritaria con poca pérdida de información, lo cual resulta beneficioso para la ejecución de los modelos al requerir menos recursos tecnológicos para el procesamiento.
- Por último, se obtienen conjuntos de datos de entrenamiento balanceados.

4. PROCESO DE ANALÍTICA

4.1 PIPELINE PRINCIPAL

El pipeline de este proyecto se ajusta muy bien a la metodología Cross-Industry Standard Process for Data Mining CRISP-DM (IBM, 2021), bajo la cual el flujo de trabajo se desarrolla como se describe a continuación:

Figura 7
Metodología CRISP-DM



Fuente: Conceptos básicos de ayuda de CRISP-DM. (IBM, 2021)

4.1.1. Comprensión del negocio

En esta primera fase el objetivo es comprender el problema de negocio al cual se pretende proponer una solución y adicionalmente, conocer el contexto de negocio para que la solución propuesta se acorde a la realidad del negocio. Es decir, entender muy bien que son operaciones inusuales y sospechosas de lavado de activos y por qué es importante para las entidades financieras contar con mecanismos efectivos en la identificación de este tipo de operaciones.

4.1.2. Comprensión de los datos

La segunda fase consiste en obtener y entender muy bien la información requerida para construcción de una solución al problema planteado, pero también puede ocurrir que a medida que se analizan los datos, se deba validar nuevamente la comprensión del problema de negocio ya que los datos suministran información adicional, muestran hipótesis o situaciones contrarias al entendimiento que se tenía hasta el momento. En esta cobra mayor relevancia el análisis exploratorio de datos de la información transaccional recopilada de los clientes.

4.1.3. Preparación de los datos

Una vez se ha comprendido el problema a solucionar y la información disponible para, se procede con la preparación de los datos según la calidad de la data disponible, acudiendo a métodos como es escalado de datos, imputación de datos, reducción de dimensionalidad, codificación de variables categóricas, entre otras técnicas de transformación de datos aplicadas en el notebook 01-EDA_operaciones_sospechosas.ipynb, el cual además contiene el desarrollo de las dos primeras fases.

4.1.4. Modelado

En esta fase se da inicio a la construcción de los modelos propuestos para la identificación de las operaciones inusuales o sospechosas de lavado de activos, para lo cual se construyen diferentes tipos de modelos y para cada uno de ellos se realizan múltiples iteraciones: inicialmente se ejecutan con los parámetros predeterminados del modelo o parámetros asignados aleatoriamente por el autor y posteriormente se realiza una optimización de dichos parámetros para la obtención del modelo con mejor desempeño según las métricas de evaluación seleccionadas. El trabajo realizado en esta fase y las siguientes está contenido en el notebook 02-Modelado_evaluacion_op_sospechosas.ipynb

4.1.5. Evaluación

Los resultados obtenidos para los modelos desarrollados se analizan para seleccionar la solución que se propondrá para el problema de negocio, el cual deberá cumplir con las condiciones definidas en el numeral 2.4 de esta monografía. Dependiendo de los resultados, se puede presentar el caso en el cual se deba retornar a la fase inicial para ajustar el pipeline o afinar alguna de las fases del mismo y así obtener un modelo que cumpla con los requerimientos del negocio.

4.1.6. Despliegue

Finalmente, cuando el modelo cumple con las condiciones establecidas por el negocio, se procede a su implementación y explotación (puesta en producción) para solucionar o abordar el problema de negocio.

4.2 PREPROCESAMIENTO

El preprocesamiento aplicado a la base de datos de manera previa al modelado (notebook 01-EDA_operaciones_sospechosas.ipynb), se hace sobre los atributos: FormaTx, ValorTx, Cliente, SaldoIniCliente, SaldoFinCliente, Beneficiario, SaldoIniBenef, SaldoFinBenef, Fraude.

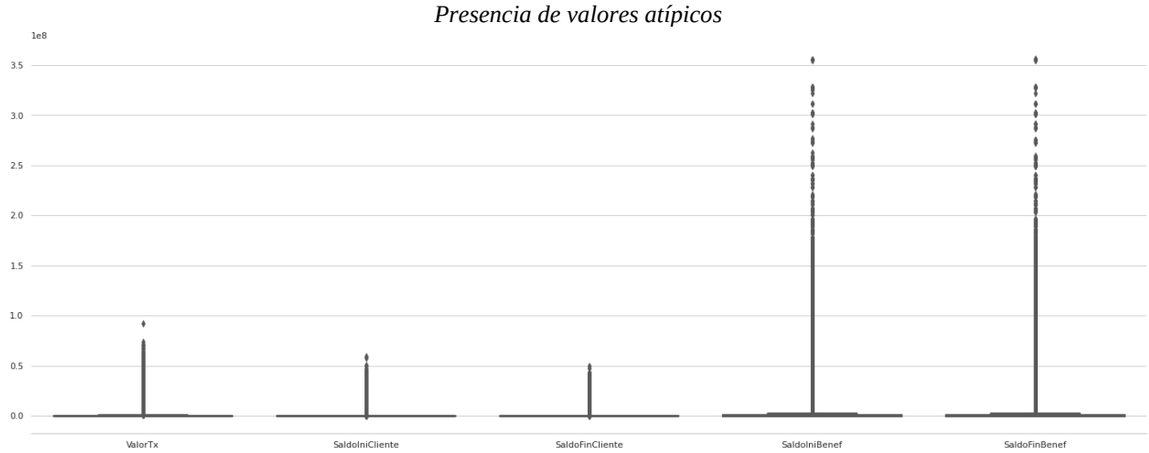
La base de datos original no cuenta con valores nulos o faltantes, razón por la cual no es necesario utilizar técnicas de imputación de datos. Las alternativas de preprocesamiento aplicadas son las siguientes:

4.2.1. Detección y eliminación de valores atípicos

Cada una de las variables numéricas tiene una alta presencia de valores atípicos, siendo el saldo inicial y saldo final del beneficiario (SaldoIniBenef, SaldoFinBenef) las variables con valores más extremos como se muestra en la siguiente gráfica:

Figura 8

Presencia de valores atípicos



Fuente: Elaboración propia.

Mediante interpolación se identifica que el valor a partir del cual los valores atípicos de SaldoIniBenef, SaldoFinBenef superan a los valores atípicos de las otras variables es desde los $60.000.000, equivalente al 0.044% de los datos disponibles y entre los cuales se encuentran sólo dos operaciones identificadas como inusuales o sospechosas, por lo cual, se procede a eliminar dichos registros.

Luego, aplicando la técnica del rango intercuartílico a estas cinco variables, se identifican los valores atípicos leves y extremos, así como la cantidad de operaciones inusuales o sospechosas relacionadas con estos valores atípicos identificando que mediante la eliminación de valores atípicos leves 1.5IQR se estarían eliminando el 63% de las operaciones inusuales o sospechosas y con la eliminación de los valores atípicos extremos 3IQR se estarían eliminando el 52.3% de estas operaciones, lo cual supone una pérdida considerable de información para el entrenamiento del modelo.

Figura 9

Identificación de valores atípicos leves y extremos

```
#rango intercuartílico de cada variable
item=['ValorTx',  'SaldoIniCliente',  'SaldoFinCliente',  'SaldoIniBenef',  'SaldoFinBenef']
dic_IQR={'Atributo':[],'minimo':[],'maximo':[]}  #Diccionario para almacenar los cuartiles de cada atributo

for i in item:
  q25,q75 = np.percentile(data[i].values, [25,75])
  iqr = q75 - q25
  dic_IQR['Atributo'].append(i)
  dic_IQR['minimo'].append(q25 - 3*iqr)
  dic_IQR['maximo'].append(q75 + 3*iqr)

#Dataframe con los rangos intercuartílicos para cada atributo
dic_IQR= pd.DataFrame(dic_IQR, index=range(len(item)))
dic_IQR
```

Fuente: Elaboración propia

Debido a la cantidad de registros de la variable objetivo que se estarían eliminando con las técnicas anteriores, es necesario aplicar el método de detección de variables atípicos no supervisado conocido como LOF (*Sklearn.neighbors.LocalOutlierFactor — Scikit-Learn 1.0.2 Documentation*, n.d.) o factor atípico local, con el cual se identifican 186.527 registros con valores atípicos que al eliminarlos representan una pérdida del 4.6% de operaciones identificadas como inusuales o sospechosas.

Figura 10

Identificación de valores atípicos con el método LOF

```
from sklearn.neighbors import LocalOutlierFactor # detección de outlers no supervisado basado en LOF
from matplotlib import pyplot

LOF = LocalOutlierFactor(n_neighbors = 5, algorithm = 'auto', metric = 'euclidean')
Filtrado = LOF.fit_predict(data[item]) # Se realiza la predicción de los datos atípicos
NOF = LOF.negative_outlier_factor_ # Detecta los valores positivos y negativos (residuos). Si los valores son grandes, entonces son valores no atípicos y por lo
                                   # Si los valores son positivos y grandes y cercanos a 1, entonces son valores atípicos. La opción negative_outlier_dactor_ cal
                                   # la media de la relación entre la densidad local de una muestra y las de sus vecinos más cercanos.

radio_outiler = (NOF.max() - NOF)/(NOF.max() - NOF.min()) # radio de detección de datos atípicos
ground_truth = np.ones(len(data[item]), dtype = int) # Se recomienda para luego comparar que datos es o no atípico (genera un vector de 1 o -1)
n_errors = (Filtrado != ground_truth).sum() # número de datos atípicos
```

```
print("Detección: ", Filtrado)
print("Factores atípicos negativos: ", NOF)
print("Número de muestras o filas con datos atípicos: ", n_errors)
```

```
Detección:  [1 1 1 ... 1 1 1]
Factores atípicos negativos:  [-1.0282857  -0.9458225  -1.0522376  ... -1.04181285 -1.0253392
 -1.30006219]
Número de muestras o filas con datos atípicos:  186527
```

Fuente: Elaboración propia.

4.2.2. Escalado de datos

Para minimizar el impacto en la escala natural de la medición de las variables de la base de datos, se aplica la técnica de escalado robusto o RobustScaler (*Sklearn.preprocessing.RobustScaler — Scikit-Learn 1.0.2 Documentation*, n.d.). En este ejercicio se aplican tres iteraciones de escalado aplicando los rangos (20,70), (25,75) y (30,80), seleccionando el rango (30,80) para la reducción de escalaridad por ofrecer el mejor resultado de reducción.

4.2.3. Codificación de las variables categóricas

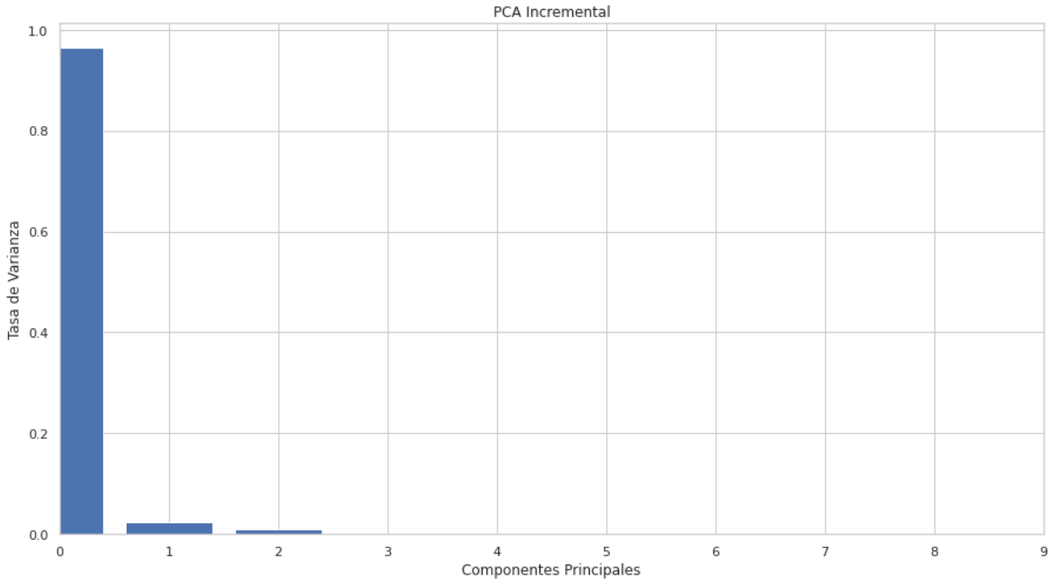
En el análisis descriptivo de los datos se evidenció que la forma de la transacción FormaTx es una variable categórica nominal que adquiere los valores: CASH_OUT, PAYMENT, CASH_IN, TRANSFER, DEBIT. Para incluirla en el modelo de clasificación, se aplica el método get_dummies de la librería Pandas con el fin de obtener variables binarias (0,1) según la forma de la transacción.

4.2.3. Reducción de dimensionalidad

Teniendo en cuenta que la codificación de las variables categóricas aumenta el número de atributos a incluir en el modelo, así como el tamaño de la base de datos, se procede a aplicar la técnica PCA incremental (*Incremental PCA — Scikit-Learn 1.0.2 Documentation*, n.d.) a los atributos utilizados para la clasificación e identificando que las primeras 3 componentes principales recopilan el 99.8% de la información original como se observa en la siguiente gráfica:

Figura 11

Importancia de las componentes principales PCA Incremental



Fuente: Elaboración propia.

Con un número de tres componentes principales como valor óptimo, se aplica la reducción de dimensionalidad utilizando el PCA clásico y ya con esto se obtiene la base de datos reducida para el entrenamiento y validación de los modelos.

4.2.4. Balanceo de datos

Por último, para compensar el desbalance de datos en las etiquetas de la variable objetivo, se aplica la técnica subsampling o submuestras en la clase mayoritaria para reducir esta clase en los conjuntos de datos de entrenamiento basados en un algoritmo similar al k-nearest neighbor para seleccionar las muestras a eliminar (Na8, 2019).

Figura 12

Balanceo de datos

```
#Aplicacíon de la técnica de submuestreo para rebalancear las clases
sub = NearMiss(n_neighbors=3)
X_train_reb, y_train_reb = sub.fit_resample(X_train, y_train)
```

Fuente: Elaboración propia.

Como resultado de la aplicación de esta técnica, los subconjuntos de entrenamiento pasaron de tener 4.297.645 registros a 10.944 con un 50% de registros correspondientes a cada una de las etiquetas de la variable objetivo. Además de subsanar el desbalance en los datos con esta técnica, también se logra reducir el tamaño de las bases de datos de entrenamiento con lo que los modelos consumirán un menor tiempo de entrenamiento.

4.3 MODELOS

Se consideraron cuatro tipos de modelos de clasificación para la identificación de operaciones inusuales o sospechosas, los cuales se entrenan y evalúan en el notebook 02-

Modelado_evaluación_op_sospechosas.ipynb. Estos modelos inicialmente se entrenan con los parámetros que tienen por defecto en sus librerías o parámetros seleccionados aleatoriamente por el autor, pero luego son optimizados con el método GridSearchCV (scikit-learn developers, n.d.) en búsqueda de los hiperparámetros que ofrecen la mejor métrica de evaluación.

A continuación, se describen cada uno de estos modelos, los cuales van aumentando su complejidad o robustez a medida que se describen:

4.3.1. Árbol de Clasificación

Es considerado el modelo base o de referencia por su sencillez y poca complejidad. Inicialmente se entrenó con los parámetros que tiene la función por defecto (Sklearn.tree.DecisionTreeClassifier — Scikit-Learn 1.0.2 Documentation, n.d.) y se le calcula el accuracy para posteriormente compararlo con la especificidad y el BACC del mismo modelo. Seguidamente se optimiza y se recalcula las métricas de desempeño.

Los parámetros iniciales y optimizados se resumen en la tabla 3 Hiperparámetros de los modelos.

4.3.2. Boosting

Este modelo aplica el meta-estimador AdaBoostClassifier y es más robusto que el anterior ya que es un ensamble de múltiples árboles de clasificación, donde cada árbol o submodelo del ensamble es ejecutado secuencialmente, aprendiendo de los errores de los árboles de clasificación precedentes y va aportando en el desempeño final del modelo (Sklearn.ensemble.AdaBoostClassifier — Scikit-Learn 1.0.2 Documentation, n.d.). Los parámetros iniciales se asignaron de manera aleatoria para tener las métricas de referencia respecto al modelo optimizado.

4.3.3. Modelo Bagging Balanceado

Estrategia de ensamble robusto de árboles de clasificación especializado en clases desbalanceadas (BalancedBaggingClassifier — Version 0.9.0, n.d.), donde n árboles de decisión son entrenados de manera aleatoria y a partir de sus predicciones individuales se compone la predicción final del modelo.

Este modelo al ser especializado para datos con clases desbalanceadas, trabaja con los datos de entrenamiento y validación originales, a diferencia de los otros modelos que se ven afectados por las clases desbalanceadas y para los cuales se utilizan los datos de entrenamiento y validación con clases balanceadas como se indica en el numeral 3.3 DATASETS.

A pesar de las bondades que ofrece este modelo para trabajar bases de datos con clases desbalanceadas, su ejecución tiene un alto costo computacional y requiere mucho más tiempo de ejecución, lo cual representa una gran desventaja frente a los otros modelos.

4.3.4. Modelo XGBoost

El modelo XGBoost es también un ensamble tipo boosting optimizado y diseñado para ser más eficiente y flexible utilizando el marco Gradient Boosting o aumento de gradiente (xgboost developers, n.d.)

A diferencia de los anteriores modelos, este no es sometido a un proceso de optimización de hiperparámetros.

Tabla 3
Hiperparámetros de los modelos

Modelo	Hiperparámetros iniciales	Hiperparámetros optimizados
Árbol de Clasificación	Por defecto	`{'max_depth': 24, 'random_state': 20}`
Boosting	`{'base_estimator__max_depth': 1, 'base_estimator__random_state': 20, 'n_estimators': 60, 'random_state': 20}`	`{'base_estimator__max_depth': 17, 'base_estimator__random_state': 20, 'n_estimators': 48, 'random_state': 20}`
Bagging Balanceado	Por defecto	`{'base_estimator__max_depth': 46, 'base_estimator__random_state': 20, 'n_estimators': 55, 'random_state': 20}`
XGBoost	`{objective='binary:logistic', booster='gbtree', seed=0}`	N/A

Fuente: Elaboración propia

4.4 MÉTRICAS

Las métricas utilizadas para evaluar el desempeño de los modelos han sido seleccionadas teniendo en cuenta los siguientes aspectos:

1. Que sean métricas recomendadas para la evaluación de modelos con datos desbalanceados y
2. Debe proporcionar información relacionada con la efectividad del modelo principalmente en la detección de operaciones sospechosas (clase 1 de la variable objetivo)

En una primera iteración y sólo como punto de referencia se calcula el accuracy o exactitud del modelo (*Sklearn.metrics.accuracy_score — Scikit-Learn 1.0.2 Documentation*, n.d.), dejando en claro que el resultado obtenido se ve fuertemente influenciado por la correcta identificación no solo de las operaciones sospechosas sino también de las operaciones normales o clase mayoritaria. El objetivo de tener esta referencia es dimensionar qué tan eficiente puede llegar a ser lo modelos que se entrenen al evaluarlos con las siguientes métricas.

La principal métrica de evaluación es la **especificidad** (Barrios, 2019)**,** la cual medirá la efectividad de los modelos en la correcta predicción o identificación de las operaciones sospechosas, dando como resultado el porcentaje de operaciones sospechosas predichas de manera correcta sobre el total de operaciones realmente sospechosas. De acuerdo a la disposición de las clases en los modelos, la forma de calcular la especificidad es de la siguiente manera:

Figura 13

Fórmula matemática de la especificidad

Especificidad (problema biclase) =

$$\frac{TN}{TN + FP}$$

=

$$\frac{Op_Sosp}{Op_Sosp + Falsas_{Op_N}orm}$$

Fuente: Elaboración propia

En la siguiente tabla se ilustra de manera más detallada cómo se hace el cálculo de la especificidad a partir de los resultados de las predicciones:

Tabla 4

Explicación de la especificidad

Especificidad	**Predicción Clase 0**	**Predicción Clase 1**
Valor Real Clase 0	TP	FN
Valor Real Clase 1	FP	TN

Fuente: Elaboración propia.

En el modelo, para el cálculo de la especificidad se define la siguiente función, aunque se puede calcular también con alguna librería de Python:

Figura 14

Cálculo de la especificidad manualmente

```
#Función para calcular la especificidad
def especificidad(confusion_matrix):
  cm= confusion_matrix[1,1]/(confusion_matrix[1,1]+confusion_matrix[1,0])
  return cm
```

Fuente: Elaboración propia.

La segunda métrica de evaluación es el **Balanced Accuracy (BACC)**, la cual calcula el promedio de sensibilidad por clase (ZACH, 2021). Esta métrica se calcula utilizando la librería de scikit learn (*Sklearn.metrics.balanced_accuracy_score — Scikit-Learn 1.0.2 Documentation*, n.d.) la cual calcula la métrica de la siguiente manera:

Figura 15

Fórmula matemática del Balanced Accuracy

Balanced Accuracy (BACC) =

$$\frac{TP}{2(TP + FN)} + \frac{TN}{2(TN + FP)}$$

Fuente: Elaboración propia.

Estas métricas, desde la perspectiva de negocio, serán aceptables al obtener por lo menos una especificidad del 70% para garantizar que la tasa de falsos positivos no sea cercana al 50% y tendrá mayor relevancia la especificidad para la elección del mejor modelo siempre y cuando no se tengan fuertes indicios de un sobreajuste o subajuste en el modelo.

5. METODOLOGÍA

5.1 BASELINE

Como modelo de referencia o baseline se tiene un árbol de decisión con sus parámetros por defecto al cual en primer lugar se le calcula el accuracy, la especificidad y el BACC. Posteriormente, con la estrategia GridSearchCV se obtiene un árbol de decisión óptimo con una profundidad de 24. Los resultados obtenidos para el modelo de referencia son los siguientes:

Figura 16
Resultados modelo base

```
   [[5472    0]
    [   0 5472]]
   [[ 595423 1244080]
    [    164    2181]]
La exactitud en Train es de: 1.0
La exactitud en Test es de: 0.32445891300476476
La especificidad en Train es de: 1.0
La especificidad en Test es de: 0.9300639658848614
El BACC en Train es de: 1.0
El BACC en Test es de: 0.6268754265247461

La especificidad en Train es de: 0.970577485380117
La especificidad en Test es de: 0.908315565031983
El BACC en Train es de: 0.985014619883041
El BACC en Test es de: 0.614837052949364
```

Fuente: Elaboración propia.

Los resultados muestran un primer modelo sobreentrenado tanto en el modelo optimizado como el estándar, con un acoracé inicial del 32%, una especificidad del 93% y un BACC del 62% sobre los datos de test. Por lo tanto, el objetivo con los otros modelos es obtener en primer lugar un modelo que no tenga sobre o subajuste y en segundo lugar que tanto el BACC como la especificidad sean altos.

5.2 VALIDACIÓN

Para la validación de los modelos se particiona la base de datos transformada en los conjuntos de datos de entrenamiento con un 70% de los datos y el conjunto de test con el 30% de datos; particiones realizadas de manera estratificada para conservar las proporciones de las etiquetas en ambos subconjuntos de datos.

Figura 17
Train/test-Split

```
#Datos de entrenamiento y prueba estratificados

X_train, X_test, y_train, y_test = train_test_split(Data_X,Data_Y,train_size=0.7, stratify=Data_Y, random_state=42)
```

Fuente: Elaboración propia

Adicionalmente, los conjuntos de train son sometidos al proceso de balanceo de datos descrito en el numeral 4.2.4. Balanceo de datos para obtener subconjuntos de entrenamiento balanceados.

Todos los modelos son entrenados con los conjuntos de datos X_train_reb, y_train_reb a excepsión del modelo Bagging Balanceado que se debe entrenar con los conjuntos de datos X_train, y_train; pero en cuanto a la validación todos utilizan los datos X_test, y_test.

La validación se realiza comparando las métricas de especificidad y BACC tanto para los datos de entrenamiento como de prueba y con ello validar la presencia de un modelo sobre ajustado, sub ajustado o aceptable.

5.3 ITERACIONES y EVOLUCIÓN

Cada una de las iteraciones ejecutadas se realizan teniendo en cuenta dos premisas: i) que el tipo de modelo a ejecutar sea más robusto o complejo respecto a las iteraciones anteriores y ii) cada modelo se ejecuta con sus hiperparámetros por defecto o seleccionados de manera aleatoria pero posteriormente se optimizan mediante la técnica GridSearchCV. Así, las iteraciones se ejecutaron en el siguiente orden:

- Iteración 1 (Baseline)

La primera iteración consiste en entrenar un árbol de decisión sencillo como se describe en el numeral 5.1. Baseline y su objetivo es tener una métrica y modelo de referencia para evaluar cualitativamente los modelos de las siguientes iteraciones.

- Iteración 2

En la segunda iteración se entrena un ensamble de árboles de decisión mediante la técnica boosting, con lo cual se busca un modelo con mejor desempeño en cuanto a la especificidad y el BACC.

- Iteración 3

Esta iteración entrena un modelo baggin balanceado, otro ensamble recomendado para bases de datos desbalanceadas y, por consiguiente, el objetivo es entrenar este modelo con los sets de datos de train-test originales y compararlo con el resultado de los otros modelos que usan los sets de datos con etiquetas balanceadas.

- Iteración 4

Se itera un modelo robusto, también basado en árboles de ensamble boosting y en la técnica descenso por gradiente.

5.4 HERRAMIENTAS

Este proyecto está desarrollado en notebooks de Jupyter los que se pueden ejecutar en Google Colab o con Anaconda.

En cuanto a las librerías ejecutadas se tienen las siguientes:

- Pandas para el procesamiento de la base de datos
- Numpy para el procesamiento de subconjuntos de datos.
- Matplotlib y Seaborn para las visualizaciones y análisis descriptivo de los datos.
- Sklearn para el preprocesamiento de los datos, modelado y evaluación de las métricas de desempeño.

- Imblearn para el resampleo y balanceo de datos de entrenamiento y modelado.
- XGBoost para el modelado de ensambles.

6. RESULTADOS

Cada uno de los modelos entrenados en las diferentes iteraciones son evaluados con las mismas métricas, es decir, la especificidad y el BACC, tanto para los conjuntos de datos de entrenamiento como de validación y a partir de dichos resultados se selecciona el mejor modelo no solo por el mayor porcentaje de especificidad, sino que sea un modelo libre de overfitting o underfitting (Tripathi, 2020).

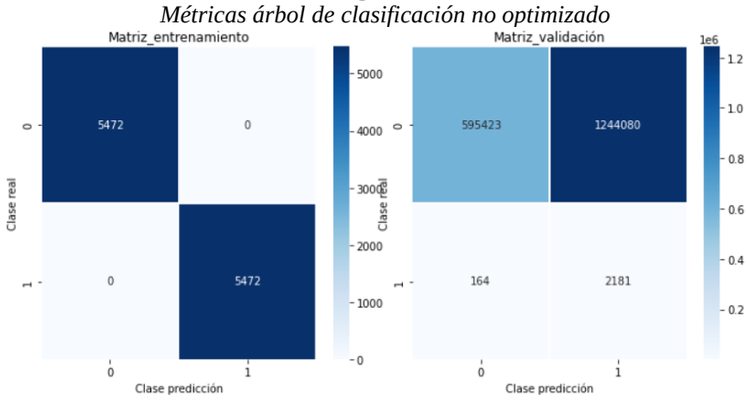
6.1 MÉTRICAS

Iteración 1 (Baseline-Árbol de Clasificación)

El primer resultado obtenido con un árbol sin optimización de los hiperparámetros y utilizando los que ofrece por defecto el modelo, proporciona una especificidad del 93% y un BACC del 63% con un modelo sobre entrenado ya que sobre los datos de entrenamiento ambas métricas son perfectas.

Figura 18

Métricas árbol de clasificación no optimizado

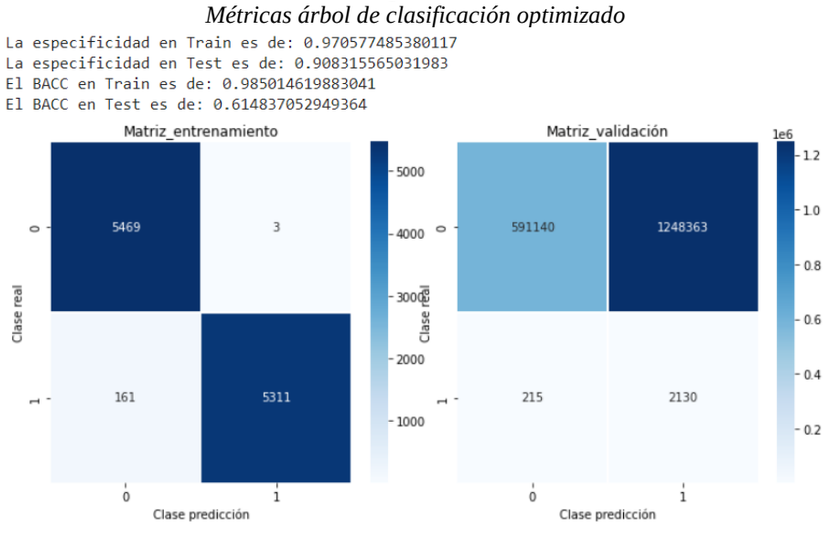


Fuente: Elaboración propia

Al optimizar este árbol de clasificación con una profundidad (max_depth) de 24, se corrige un poco el sobre entrenamiento a costa de castigar un poco la especificidad y el BACC.

Figura 19

Métricas árbol de clasificación optimizado



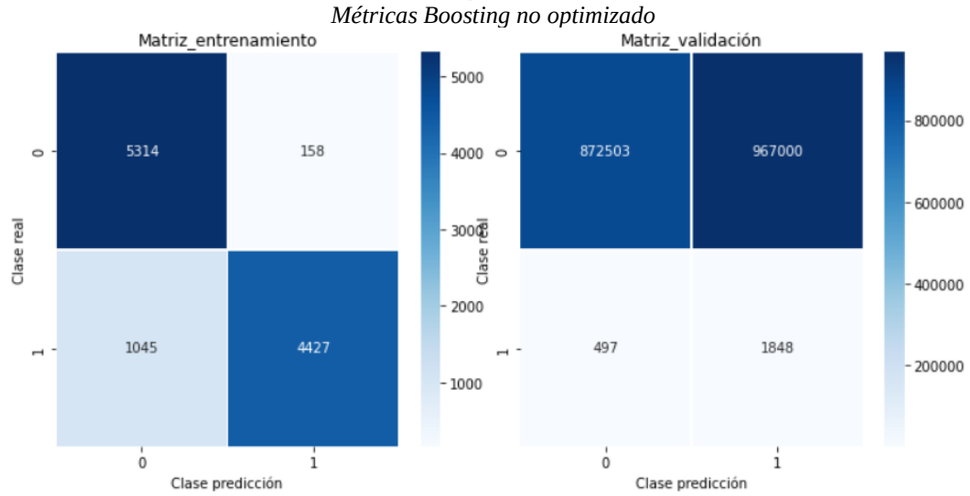
Fuente: Elaboración propia

Iteración 2- Ensamble Boosting

Este modelo sin optimización ofrece una especificidad del 78,8% y un BACC del 63,1 % en los datos de prueba frente a un 80,9% y 89,9% en los datos entrenamiento, por lo cual, con base a estos resultados, parece ser un buen modelo al proporcionar buenos resultados y no tener indicios de sobre o sub-entrenamiento.

Figura 20

Métricas Boosting no optimizado

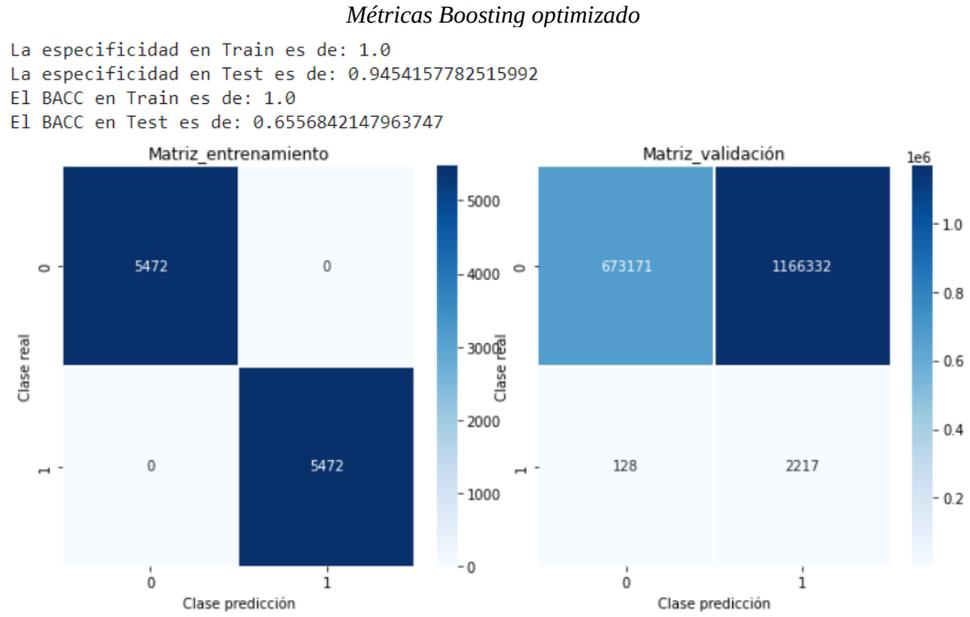


Fuente: Elaboración propia

Al realizar la optimización de los hiperparámetros de este ensamble, las métricas mejoran, pero el modelo se sobreentrena.

Figura 21

Métricas Boosting optimizado

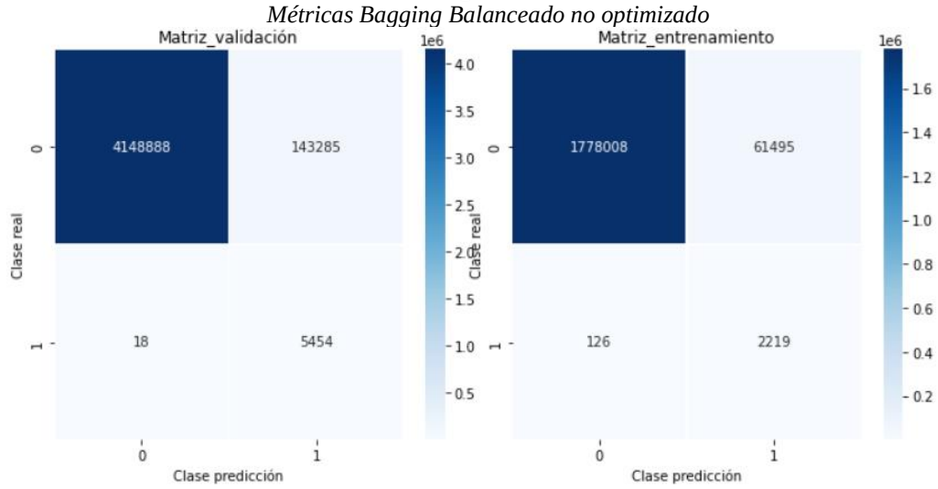


Fuente: Elaboración propia.

Iteración 3- Bagging Balanceado

Figura 22

Métricas Bagging Balanceado no optimizado




```
La especificidad en Train es de: 0.9967105263157895
La especificidad en Test es de: 0.9462686567164179
El BACC en Train es de: 0.9816638343641346
El BACC en Test es de: 0.9564192156348266
```

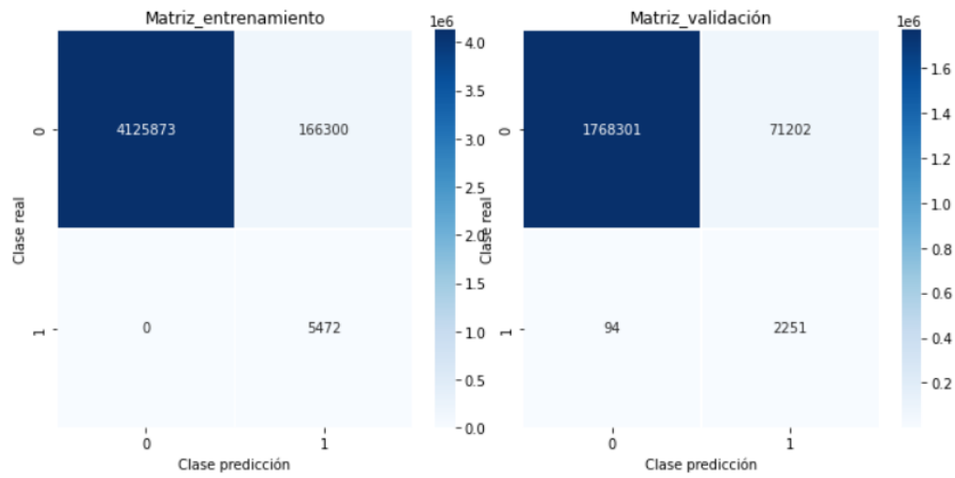
Fuente: Elaboración propia

Las métricas de este modelo con los parámetros que trae la función por defecto son muy buenos, pues tanto en los conjuntos de datos de entrenamiento como de prueba son superiores al 94% y la brecha entre el BACC y la especificidad es muy pequeña. Tampoco pareciera haber sobre entrenamiento ya que las métricas de desempeño en ambos conjuntos de datos son muy similares. Al igual que en la iteración 2, al optimizar los parámetros de este modelo, el mismo cae en sobre entrenamiento.

Figura 23

Métricas Bagging Balanceado optimizado

```
La especificidad en Train es de: 1.0
La especificidad en Test es de: 0.9599147121535181
El BACC en Train es de: 0.9806275282939434
El BACC en Test es de: 0.9606037589366619
```



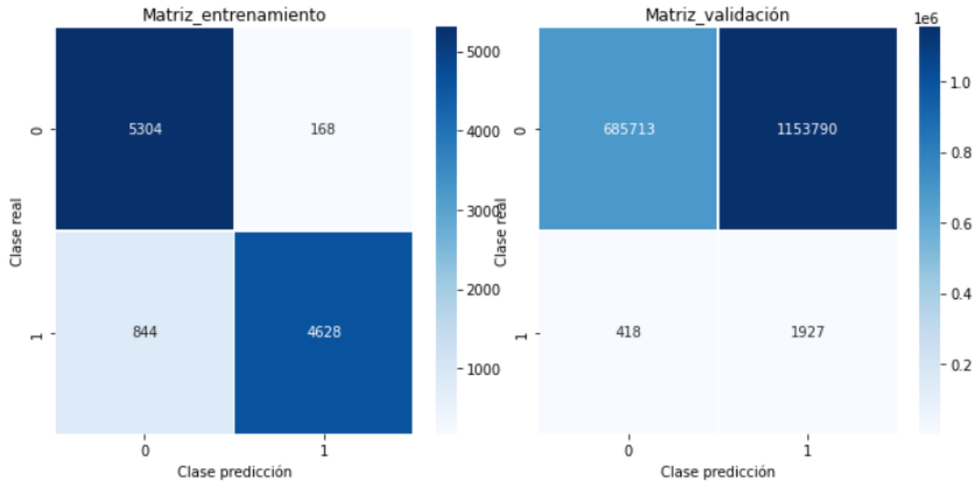
Fuente: Elaboración propia.

Iteración 4- Modelo XGBoost

Los resultados de este modelo son comparativos con los de la iteración anterior ya que no tiene indicios de estar sobre entrenado o sub entrenado, obteniendo una especificidad del 84,6% y 82,2% con los datos de entrenamiento y de prueba respectivamente y un BACC del 90,7% y 59,7% respectivamente.

Figura 24

Métricas del modelo XGBoost



Fuente: Elaboración propia.

6.2 EVALUACIÓN CUALITATIVA

Al comparar las métricas obtenidas con las métricas de negocio requeridas, estos modelos sólo logran satisfacer la condición de alcanzar una especificidad o BACC mayor al 70% en la identificación de las operaciones inusuales o sospechosas reales, ya que generan altos porcentajes (superiores al 50%) de operaciones normales clasificadas como operaciones sospechosas por el modelo (baja precisión). Esto último resulta costoso operativamente para la entidad financiera en la gestión de dichas falsas operaciones sospechosas.

Por otra parte, el orden en que se ejecutaron los modelos permite observar, cómo para este tipo de problemas, los modelos más sencillos tienden a caer en sobre entrenamiento más fácil que los modelos robustos y a su vez al optimizar los modelos, también aumenta la posibilidad de obtener un sobre entrenamiento del modelo

Por lo tanto, el modelo más recomendado para las entidades financieras en la identificación de operaciones inusuales o sospechosas es el Bagging Balanceado, ya que ofrece una especificidad y BACC de hasta el 96% en la identificación de las operaciones inusuales o sospechosas reales y la menor tasa de falsas operaciones inusuales o sospechosas (precisión del 3% aproximadamente).

6.3 CONSIDERACIONES DE PRODUCCIÓN

Para explotar o poner en producción el modelo para la detección de operaciones inusuales o sospechosas de lavado de activos, se debe tener claro es la oportunidad con la cual se desean conocer los resultados, es decir, si se desea realizar una labor preventiva antes de que un cliente finalice la transacción o si la labor será detectiva o de manera posterior a la realización de las transacciones. De esto depende la integración de la herramienta o plataforma en la cual se aloje el modelo con las respectivas fuentes de información transaccional.

Una vez establecida la conexión entre las fuentes de información con la herramienta que ejecutará el modelo, se debe configurar dicha herramienta de tal manera que el modelo sea inicializado cuando se vaya a realizar una transacción si es con fines detectivos, para que el modelo se ejecute automáticamente cada periodo de tiempo definido por la entidad financiera.

También se debe establecer cómo entregará el resultado el modelo, si mediante una señal de alerta a alguna persona, a un aplicativo o si dicho resultado será guardado en una base de datos.

Esta es una descripción a grandes rasgos de cómo sería la puesta en producción de este modelo y su complejidad o forma de hacerlo dependerá del objetivo y de las herramientas con las cuales cuente la entidad financiera. Durante el tiempo de vida o de explotación del modelo, se debe hacer seguimiento al desempeño del mismo, pues cuando se tenga evidencia de que el número de falsos positivos se está incrementando o el número de operaciones inusuales o sospechosas no están siendo clasificadas de manera correcta, se debe calibrar o reentrenar el modelo con información actualizada.

7. CONCLUSIONES

Los resultados presentados en esta monografía evidencian el desafío que implica el desarrollo de modelos de clasificación con data que presenta altos niveles de desbalance de clases, como ocurre en los casos de identificación de operaciones de fraude, de lavado de activos, corrupción, entre otros.

Este tipo de problemas requieren de métodos o técnicas robustas y especializadas en el procesamiento de datos desbalanceados. Como se pudo observar, el Bagging Balanceado es el modelo con mejor desempeño y no requiere un tratamiento previo para compensar el desbalance en los conjuntos de datos de entrenamiento como se hizo para los otros modelos. Esto se debe a que el Bagging Balanceado incorpora el proceso de equilibrar las clases en el conjunto de entrenamiento.

En cuanto al costo de implementación de estos modelos, también resulta un poco alto ya que el consumo computacional es importante, esto en función de la complejidad del modelo y la cantidad de información a procesar, pues justamente, a medida que se probaba con un modelo más robusto, se requería de más tiempo y más memoria para la ejecución y entrenamiento de los mismos.

La elección de las métricas de desempeño es fundamental y para ello se debe tener muy claro el problema y el objetivo del proyecto a desarrollar, pues generalmente se tienen disponibles diferentes posibilidades para evaluar los modelos y en algunos casos ofrecen interpretaciones diferentes, por lo cual, la métrica debe tener la capacidad de medir el cumplimiento de los objetivos del proyecto. Por ejemplo, para el problema abordado en esta monografía, es más crítico para las entidades financieras la no identificación de una operación inusual o sospechosa de lavado de activos que la gestión de falsos positivos u operaciones normales que son clasificadas como sospechosas, sin importar lo que esto último pueda costarle.

8. REFERENCIAS

BalancedBaggingClassifier — Version 0.9.0. (n.d.). Imbalanced Learn. Retrieved April 13, 2022, from https://imbalanced-learn.org/stable/references/generated/imblearn.ensemble.BalancedBaggingClassifier.html

Barrios, J. I. (2019, 07 26). *La matriz de confusión y sus métricas – Inteligencia Artificial –*. Juan Barrios. Retrieved March 27, 2022, from https://www.juanbarrios.com/la-matriz-de-confusion-y-sus-metricas/

Cómo la asimetría y la curtosis afectan la distribución - Minitab. (n.d.). Support. Retrieved March 27, 2022, from https://support.minitab.com/es-mx/minitab/18/help-and-how-to/statistics/basic-statistics/supporting-topics/data-concepts/how-skewness-and-kurtosis-affect-your-distribution/

ESEBAMEN, K. (n.d.). *Python · Synthetic Financial Datasets For Fraud Detection*. AML detection. https://www.kaggle.com/code/x09072993/aml-detection/data

Gonzalez, L. (2019, June 14). *Conjunto de datos desbalanceado - Aprende IA*. Aprende IA. Retrieved March 27, 2022, from https://aprendeia.com/conjunto-de-datos-desbalanceado/

IBM. (2021, 08 17). *Conceptos básicos de ayuda de CRISP-DM*. IBM. Retrieved April 10, 2022, from https://www.ibm.com/docs/es/spss-modeler/SaaS?topic=dm-crisp-help-overview

Incremental PCA — scikit-learn 1.0.2 documentation. (n.d.). Scikit-learn. Retrieved April 23, 2022, from https://scikit-learn.org/stable/auto_examples/decomposition/plot_incremental_pca.html

Na8. (2019, May 16). *Clasificación con datos desbalanceados*. Aprende Machine Learning. Retrieved March 27, 2022, from https://www.aprendemachinelearning.com/clasificacion-con-datos-desbalanceados/

scikit-learn developers. (n.d.). *sklearn.model_selection.GridSearchCV — scikit-learn 1.0.2 documentation*. Scikit-learn. Retrieved April 13, 2022, from https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

Sierra Fajardo, O. (2021, February 24). *El secreto bancario, al descubierto*. Ámbito Jurídico. Retrieved April 9, 2022, from

https://www.ambitojuridico.com/noticias/especiales/penal/el-secreto-bancario-al-descubierto

sklearn.ensemble.AdaBoostClassifier — scikit-learn 1.0.2 documentation. (n.d.). Scikit-learn. Retrieved April 13, 2022, from https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html

sklearn.metrics.accuracy_score — scikit-learn 1.0.2 documentation. (n.d.). Scikit-learn. Retrieved March 27, 2022, from https://scikit-learn.org/stable/modules/generated/sklearn.metrics.accuracy_score.html

sklearn.metrics.balanced_accuracy_score — scikit-learn 1.0.2 documentation. (n.d.). Scikit-learn. Retrieved March 27, 2022, from https://scikit-learn.org/stable/modules/generated/sklearn.metrics.balanced_accuracy_score.html

sklearn.neighbors.LocalOutlierFactor — scikit-learn 1.0.2 documentation. (n.d.). Scikit-learn. Retrieved April 23, 2022, from https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.LocalOutlierFactor.html

sklearn.preprocessing.RobustScaler — scikit-learn 1.0.2 documentation. (n.d.). Scikit-learn. Retrieved April 23, 2022, from https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.RobustScaler.html

sklearn.tree.DecisionTreeClassifier — scikit-learn 1.0.2 documentation. (n.d.). Scikit-learn. Retrieved April 13, 2022, from https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html

Superintendencia Financiera de Colombia. (2020). Capítulo IV: Instrucciones relativas a las administración del riesgo de lavado de activos y de la financiación del terrorismo. In *Circular Externa 027*.

3.1. Cross-validation: evaluating estimator performance — scikit-learn 1.0.2 documentation. (n.d.). Scikit-learn. Retrieved March 27, 2022, from https://scikit-learn.org/stable/modules/cross_validation.html#stratification

Tripathi, M. (2020, June 13). *Underfitting and Overfitting in Machine Learning*. Data Science Foundation. Retrieved April 30, 2022, from https://datascience.foundation/sciencewhitepaper/underfitting-and-overfitting-in-machine-learning

Unidad de Inteligencia y Análisis Financiero. (2018, noviembre 26). *Lavado de Activos - Unidad de Información y Análisis Financiero UIAF*. UIAF. Retrieved April 9, 2022, from https://www.uiaf.gov.co/sistema_nacional_ala_cft/lavado_activos_financiacion_29271/lavado_activos

xgboost developers. (n.d.). XGBoost Documentation — xgboost 1.5.2 documentation. Retrieved April 13, 2022, from https://xgboost.readthedocs.io/en/stable/

ZACH. (2021, October 7). *How to Calculate Balanced Accuracy in Python Using sklearn - Statology*. - Statology. Retrieved March 27, 2022, from https://www.statology.org/balanced-accuracy-python-sklearn/