

**Análisis y Predicción de Fraude**

**AWS**

**Informe Final**

**Mayo de 2023**

**Presentado por:**

Valentín Altamirano

Agustín Borges

Mateo Patrone

**Docentes:**

Claudio Somma, Ingeniero en Computación

Lucía Etchecopar, Ingeniera en Informática

Victoria Seoane, Ingeniera Telemática

**Índice**

[**Introducción** 4](#_Toc135816592)

[**Diseño de la solución** 5](#_Toc135816593)

[**Diagrama de servicios utilizados** 5](#_Toc135816594)

[**Diagrama de red** 6](#_Toc135816595)

[**Estructura de la base de datos** 6](#_Toc135816596)

[**Implementación y Evaluación de la Solución** 6](#_Toc135816597)

[**Estructura y contenido de los tableros** 6](#_Toc135816598)

[**Fecha y Hora** 7](#_Toc135816599)

[**Distribución** 7](#_Toc135816600)

[**Género** 8](#_Toc135816601)

[**Categoría** 9](#_Toc135816602)

[**Ubicación** 9](#_Toc135816603)

[**Invocación de la API** 10](#_Toc135816604)

[**Lecciones aprendidas y posibles mejoras futuras** 11](#_Toc135816605)

[**Anexo** 13](#_Toc135816606)

[**1.0 Diagrama de servicios utilizados** 13](#_Toc135816607)

[**2.0 Diagrama de red** 13](#_Toc135816608)

[**3.0 Estructura de la base de datos** 14](#_Toc135816609)

[**4.0 Matriz de confusión modelo inicial** 15](#_Toc135816610)

[**5.0 Matriz de confusión modelo final** 15](#_Toc135816611)

[**6.0 Repositorio de GitHub** 15](#_Toc135816612)

# **Introducción**

En la era digital actual, el fraude con tarjetas de crédito se ha convertido en una preocupación creciente para bancos, empresas y consumidores. La prevalencia y sofisticación de estos actos fraudulentos no sólo generan pérdidas financieras significativas, sino que también erosionan la confianza en los sistemas de pago digitales. Ante esta creciente amenaza, es esencial emplear un enfoque proactivo y basado en datos para la detección y prevención del fraude. En este informe, presentamos un proyecto en el que hemos utilizado diversas herramientas y servicios de Amazon Web Services (AWS) para analizar un conjunto de datos de transacciones con tarjetas de crédito y desarrollar un sistema capaz de detectar patrones de fraude.

Nuestra propuesta de solución se basa en un análisis simple pero exhaustivo y sistemático de los datos disponibles, buscando patrones y características distintivas de las transacciones fraudulentas. Exploramos diversas dimensiones de los datos, incluyendo la ubicación geográfica, el número y monto de las transacciones, el horario en que se realizan, la categoría de la transacción, el género del titular de la tarjeta, entre otros factores (columnas de la tabla que creamos a partir del CSV). La identificación de estos patrones es fundamental para comprender el comportamiento fraudulento y diseñar estrategias de prevención eficaces.

Una parte crucial de nuestro trabajo fue la utilización de Amazon QuickSight para el análisis exploratorio de los datos y la visualización de los resultados. QuickSight es una herramienta poderosa que nos permitió obtener insights profundos y visualizar las tendencias y patrones en nuestros datos de manera clara y concisa. Este enfoque basado en visualizaciones permitió no solo un análisis eficaz de los datos, sino también una comunicación efectiva de los resultados y descubrimientos a las partes interesadas, que era lo que nos interesaba.

Mediante este análisis, hemos construido un modelo de detección de fraudes que, en última instancia, busca proporcionar a los bancos y otras entidades interesadas una herramienta valiosa para combatir el fraude, empezando por la detección de las mismas. Este modelo, entrenado con un conjunto diverso de características de las transacciones, puede asignar a cada nueva transacción una probabilidad de ser fraudulenta. Este enfoque basado en probabilidades permite a las instituciones financieras tomar decisiones informadas y actuar de manera proactiva para prevenir el fraude.

Es importante destacar que este proyecto no sólo involucró un análisis de datos detallado, sino también la construcción de una infraestructura en AWS que permitiera un manejo eficiente de los datos y la implementación del modelo. Para ello, hemos utilizado servicios de AWS como S3 para el almacenamiento de datos, RDS para la gestión de bases de datos, Lambda para la ejecución de código sin servidor y la invocación de nuestra API, y VPC para la creación de una red virtual segura.

En el curso de este informe, proporcionaremos detalles sobre los servicios utilizados, los desafíos encontrados, cómo los abordamos, y las lecciones aprendidas de este proyecto. A través de este trabajo, buscamos demostrar cómo la combinación de técnicas de análisis de datos y servicios en la nube puede dar lugar a soluciones poderosas y escalables para problemas críticos como el fraude con tarjetas de crédito.

# **Diseño de la solución**

## **Diagrama de servicios utilizados**

Para gestionar la creación y administración de nuestros recursos en AWS, elegimos usar CloudFormation, una herramienta muy eficiente que nos permitió describir y aprovisionar todos los recursos de AWS necesarios.

El primer recurso que creamos fue un **Bucket** en **S3**, para el cual diseñamos una pila en CloudFormation. Este Bucket de S3 se utilizó para almacenar y administrar un archivo CSV, que contenía datos de aproximadamente 550,000 transacciones bancarias. Cada registro en este archivo tenía información sobre la categoría de la transacción, la fecha y hora, la cantidad de dinero implicada, la latitud y longitud tanto del emisor como del receptor de la transacción, y si la transacción resultó ser un fraude, entre otros detalles. Estos datos fueron cruciales para el análisis que realizamos.

Además, también utilizamos el Bucket de S3 para guardar un archivo de tipo Joblib que contiene un modelo de regresión logística que desarrollamos para predecir la probabilidad de que una transacción sea fraudulenta. Este modelo se utilizó más tarde en una función Lambda, que estaba conectada a una API para consultas sobre la probabilidad de fraude en transacciones futuras.

Para un almacenamiento eficiente y seguro de nuestra base de datos, creamos una **VPC (Virtual Private Cloud)** a través de otra pila en CloudFormation. Esta VPC nos proporcionó un data center virtual óptimo para albergar nuestra base de datos, lo que nos permitió organizar y manejar nuestros datos de manera más eficiente.

Nuestra solución también hizo un uso extensivo de **AWS Lambda Function**. Para cada función Lambda que creamos, generamos una pila separada en CloudFormation.

Debido a que el coste del proyecto empezó a incrementarse con el tiempo, decidimos optimizar nuestros gastos desarrollando dos funciones Lambda. Una función se encargaba de activar la instancia de RDS y la otra de apagarla, y ambas eran activadas una vez al día por un evento de **Amazon EventBridge**. Programamos estos eventos en la sección de programaciones para que se ejecutaran periódicamente cada 24 horas.

Posteriormente, desarrollamos una función Lambda adicional, cuya principal funcionalidad era cargar los datos procesados en una instancia de **RDS** (MySQL) para su almacenamiento y futuro análisis en **QuickSight**. Para lograr esto, utilizamos la biblioteca PyMySQL de Python. Sin embargo, como AWS Lambda no viene con PyMySQL preinstalado, tuvimos que crear una layer (capa) separada para esta función que incluía sus dependencias empaquetadas en un archivo zip.

Con el fin de mantener un seguimiento de la carga de datos y su eficacia, utilizamos DBeaver, que nos proporcionó una visión clara de nuestros datos almacenados en la base de datos.

Finalmente, la última función Lambda que creamos constaba de tres partes. La primera parte estaba destinada a abrir el modelo de regresión logística almacenado en un archivo Joblib en el Bucket. Por otra parte, mediante la invocación de una API a través de Postman, la función recibiría datos acerca de una transacción. Por último, pero no menos importante, este conjunto de datos es procesado por nuestro modelo de regresión logística. Sin embargo, nos enfrentamos a desafíos debido a las limitaciones de AWS Lambda. Específicamente, tuvimos que crear una capa adicional para incluir la librería Joblib, al igual que habíamos hecho previamente para PyMySQL. A su vez, necesitábamos la librería sklearn, requerida por el modelo. Sin embargo, esta librería excedía el límite de tamaño permitido de 50 MB para las capas de AWS Lambda cuando se comprime en un archivo zip. Para superar esta limitación, recurrimos a una capa preexistente alojada en GitHub que incluye la librería sklearn. Con estos recursos a nuestra disposición, el modelo es capaz de evaluar las características de la transacción proporcionada, retornando a la API la probabilidad de que dicha transacción sea fraudulenta. Para realizar un seguimiento de las invocaciones de la función Lambda y garantizar su correcto funcionamiento, utilizamos **AWS CloudWatch**, que nos permite visualizar de manera eficiente los registros de la función cada vez que se invoca a través de la API.

## **Diagrama de red**

En la sección 2.0 del anexo se puede apreciar el diagrama de la red implementada. Optamos por las availability zones us-east-1a y us-east-1b porque son las que tienen mejor flujo de información desde Uruguay. Esta estructura le da robustez y aumenta la disponibilidad de la misma. Al crear la instancia de RDS publica, conectamos la misma a las dos subredes públicas de nuestra VPC.

## **Estructura de la base de datos**

Para modelar la estructura de nuestra base de datos, optamos por consolidar toda la información contenida en el archivo CSV original en una única tabla, adecuadamente adaptada para el entorno MySQL. Esta tabla, denominada “transactions”, almacena múltiples parámetros relacionados con las transacciones individuales, cada uno con su tipo de dato correspondiente. En el anexo, bajo la sección 3.0 se puede apreciar el contenido de la tabla.

La tabla 'transactions' incluye desde datos básicos de transacciones como el número de la tarjeta de crédito (cc\_num), la fecha y hora de la transacción (trans\_date\_trans\_time), el comerciante (merchant), la categoría de la transacción (category), y la cantidad (amt), hasta datos demográficos y de ubicación más personales, como nombre (first\_name, last\_name), género (gender), dirección (street, city, state, zip), la población de la ciudad (city\_pop), ocupación (job), fecha de nacimiento (dob), y coordenadas geográficas (latitude, longitude, merch\_lat, merch\_long).

Además, la tabla incorpora dos elementos críticos: el identificador de transacción (trans\_num), que es único para cada transacción, y una variable de salida que indica si la transacción es fraudulenta o no (is\_fraud). Este último es especialmente vital ya que constituye el objetivo en nuestro modelo de detección de fraude.

Aunque nuestra solución actual se basa en una única tabla, reconocemos que esta estructura podría presentar algunos desafíos y limitaciones, especialmente en términos de flexibilidad y escalabilidad. Como se detalla en la sección de lecciones aprendidas y mejoras futuras, una posible mejora sería evolucionar hacia un diseño más normalizado de la base de datos, con varias tablas relacionadas que podrían proporcionar una gestión más eficiente de los datos.

# **Implementación y Evaluación de la Solución**

## **Estructura y contenido de los tableros**

En QuickSight nuestro principal objetivo es analizar los datos que tomamos del dataset y comunicar las relaciones entre los datos, las tendencias y patrones de los mismos. En la primera parte representamos los indicadores clave generales en base a los datos. Esto brinda un contexto al usuario y le presenta un paneo de los datos más generales involucrados en las transacciones históricas registradas.

El análisis lo dividimos en hojas. Cada una de ellas con uno o dos pares de eventos visuales con un propósito en común.

### **Fecha y Hora**

La primera de ellas está basada en la fecha y hora de las transacciones en general y las fraudulentas.

El primer elemento visual es un gráfico combinado de barras agrupadas que muestra las transacciones y transacciones fraudulentas semanales dentro del período que se tienen datos. En celeste las barras representan el número acumulado en cada semana de transacciones tanto fraudulentas como no fraudulentas y la línea azul representa el mismo concepto, pero filtrado por transacciones fraudulentas. Este gráfico evalúa el volumen de transacciones a lo largo del segundo semestre del año y si los fraudes acompañan el ritmo de las transacciones o si presentan otro comportamiento.

El segundo par de elementos visuales son mapas térmicos donde las filas son los días del mes que se efectúan las transacciones y las columnas son las horas de 0 a 23. Si bien cada punto representa el acumulado de transacciones de un día (de 1 a 31) y de hora (de 0 23) en específico, lo que explicita este gráfico es el cuándo se efectúa el mayor volumen de transacciones durante un día y durante el mes. Separamos en dos mapas térmicos diferenciando otra vez aquellas transacciones fraudulentas y las transacciones totales.

Se puede observar un claro patrón en el flujo de transacciones realizadas y fraudulentas. Existen transacciones en todos los horarios, pero un mayor volumen a partir del mediodía hasta las 23hs. Vemos que los fraudes no acompasan el volumen de transacciones, sino que responden a otro patrón. Este fenómeno podría estar relacionado con una serie de factores.

Este patrón de actividad fraudulenta nocturna puede deberse a las tácticas de los perpetradores de fraude. Podrían preferir operar durante la noche para aprovechar las horas en las que los usuarios y los sistemas de vigilancia de las instituciones financieras podrían estar menos activos o alerta. La relativa calma de este período podría proporcionar a los defraudadores una ventana de oportunidad para llevar a cabo actividades fraudulentas con un menor riesgo de detección inmediata.

### **Distribución**

Se presenta inicialmente pequeños eventos visuales que proporcionan un contexto de la cantidad de dinero de las transacciones en general y las transacciones fraudulentas. Se presenta el mínimo, máximo y promedio de transacciones totales y luego filtrado por fraudulentas.

Debajo presentamos un par de histogramas que explicitan la distribución de la cantidad de dinero de las transacciones registradas. Cada “papelera” o barra acumula las transacciones dentro de rangos de 10 dólares de 0 a 1.000.

En cuanto a las transacciones totales, se observa que la inmensa mayoría (más del 98%) son de cantidades menores a 250 dólares, siendo la transacción media de 69 dólares. Esto parece ser coherente con las expectativas, ya que las transacciones del día a día suelen ser de montos más bajos. Este patrón se refleja claramente en el histograma, con una alta concentración de transacciones en las primeras "papeleras" que corresponden a montos más bajos.

Sin embargo, la situación cambia drásticamente cuando se consideran únicamente las transacciones fraudulentas. Aunque los picos más altos en el histograma de transacciones fraudulentas también se encuentran en los intervalos de 0 a 10 y de 10 a 20 dólares, es notable que más del 85% del número total de transacciones fraudulentas se da a partir de los 200 dólares. Además, la transacción fraudulenta promedio es considerablemente mayor que la transacción promedio general, siendo de 528 dólares.

Esto sugiere que los defraudadores tienden a realizar transacciones de montos mayores, con el incentivo de maximizar sus ganancias antes de que el fraude sea detectado. Sin embargo, también parecen intentar transacciones fraudulentas chicas, probablemente como una forma de "probar" las tarjetas de crédito antes de realizar transacciones de montos más grandes.

### **Género**

Se presentan tres pares de elementos visuales. El primer par representa en dos gráficos circulares el dato crudo del acumulado de transacciones que efectúan hombres y mujeres. Junto a él el mismo concepto, pero filtrado por transacciones fraudulentas.

El segundo par de elementos visuales son gráficos de barras horizontales que acumulan el número transacciones efectuadas por cada categoría discriminadas por el género de titulares de la tarjeta: hombres y mujeres. El tercero en lugar de acumular la cantidad de transacciones acumula la cantidad de dinero involucradas en las transacciones. Cada uno de estos pares presenta primero la totalidad de las transacciones registradas y segundo filtrado por transacciones fraudulentas

De los gráficos circulares, podemos ver que el porcentaje de transacciones entre hombres y mujeres está muy cerca con las mujeres que realizan ligeramente más transacciones en general y fraudulentas, representando el 55% y el 54% respectivamente. Esta distribución equilibrada se refleja en la cantidad de transacciones realizadas por hombres y mujeres, así como en las transacciones fraudulentas. Esto indica que el género por sí solo no es un indicador significativo de fraude, y no hay un sesgo considerable hacia un género.

Al mirar los gráficos de barras horizontales de las transacciones totales y fraudulentas desglosadas por categoría, observamos algunas diferencias notables. En el caso de las transacciones totales, las mujeres tienden a realizar más transacciones en casi todas las categorías, con "gas\_transport", "grocery\_pos" y "shopping\_pos" liderando. Los hombres, por otro lado, superan a las mujeres en las categorías "home" y "grocery\_net".

Sin embargo, cuando analizamos las transacciones fraudulentas, observamos que la categoría "shopping\_net" es la más afectada por el fraude en ambos géneros. También es notable el hecho de que las transacciones fraudulentas en la categoría "travel" son significativamente más altas para las mujeres que para los hombres.

El análisis del monto de las transacciones proporciona una visión interesante. Para las transacciones totales, los hombres gastan más en "travel", mientras que las mujeres gastan más en "grocery\_pos" y "shopping\_pos". En cuanto a las transacciones fraudulentas, las categorías "shopping\_pos" y "shopping\_net" tienen los montos más altos en ambos géneros, lo que indica que estos son los segmentos preferidos para el fraude.

Aunque el género por sí solo no parece ser un factor de riesgo significativo para el fraude, los datos sugieren que ciertas categorías son más propensas al fraude que otras, independientemente del género del titular de la tarjeta. Las categorías "shopping\_net" y "shopping\_pos" son particularmente vulnerables, lo que indica la necesidad de medidas de seguridad adicionales en estas áreas.

### **Categoría**

En esta hoja se presenta un par de elementos visuales con el objetivo de representar la diferencia de fraudes entre las distintas categorías.

Ambos son gráficos de barras horizontales. El primero muestra en orden descendente el número de transacciones acumuladas de transacciones totales en celeste y en azul la cantidad de transacciones fraudulentas. La segunda muestra de la misma forma, pero en lugar de acumular el número de transacciones las barras acumulan la suma de dinero de las transacciones.

El análisis de estos gráficos revela que, a pesar de que las categorías "grocery\_pos", "shopping\_pos" y "gas\_transport" son las que manejan la mayor cantidad de dinero, las categorías con más actividad fraudulenta son "shopping\_net" y "misc\_net". Esta discrepancia se puede atribuir a las diferencias en la seguridad de las transacciones entre estas categorías.

Por ejemplo, una transacción de compra en línea, que se categorizaría como "shopping\_net", se puede realizar con solo los datos de la tarjeta y del titular en caso de que no poseer tokens u otros mecanismos de seguridad adicionales. En comparación, una transacción realizada en persona en una tienda física ("shopping\_pos") requiere la tarjeta física y a menudo también un código PIN. Por lo tanto, es más fácil para los delincuentes realizar fraudes en la categoría "shopping\_net".

Además, a pesar de que "shopping\_net" no tiene el mayor número de transacciones, se encuentra entre las categorías que manejan más dinero, situándose en la cuarta posición. Esta combinación de seguridad laxa y altos valores de transacción hace que "shopping\_net" sea un objetivo atractivo para el fraude. De ahí que es el número uno en transacciones fraudulentas tanto en número como en cantidad de dinero. La categoría "misc\_net" presenta características similares, aunque en una escala menor.

Esta hoja proporciona una visión útil de las tendencias de fraude en diferentes categorías de transacciones, destacando la importancia de fortalecer la seguridad, especialmente en aquellas categorías más vulnerables al fraude y en el caso de los bancos poner mayor atención aquellas con menor seguridad.

### **Ubicación**

En la hoja que representa el análisis basado en la ubicación de las transacciones, presentamos dos pares de mapas para observar distintas relaciones.

El primer par de mapas utilizamos los llamados “filled maps” para identificar en clave de intensidad de colores el volumen de transacciones que se efectúan por estado del titular de la tarjeta y luego como en general hicimos, analizando las transacciones fraudulentas y también por volumen de dinero involucrado en las transacciones.

Para hacerlo más interactivo, antes de cada mapa incluimos unos parámetros para poder optar por visualizar el acumulado de cantidad de transacciones o el acumulado de dinero.

El segundo par de mapas representa la ubicación geográfica del titular de la tarjeta sobre la cual se hicieron transacciones fraudulentas y la ubicación geográfica de los comerciantes donde se realizaron transacciones fraudulentas.

Se utilizó en este caso el elemento visual llamado “puntos en el mapa” para representar en forma de puntos la ubicación tomando como dato geoespacial la latitud y longitud de la transacción para uno y lo mismo para el comerciante en el otro. En el caso de la ubicación de los titulares de la tarjeta involucrada en el fraude utilizamos el estilo de punto básico y que el tamaño del punto sea de cuerdo al número de transacciones. Esto es porque varias transacciones fraudulentas coincidían entre sí en la ubicación del titular de la tarjeta. EN el caso de la ubicación de los comerciantes utilizamos puntos de clúster. Esto es porque todas las ubicaciones eran distintas entre sí y con este estilo de punto permite tener una noción de la cantidad de ellos es distintas zonas del territorio americano.

Sobre los datos es importante notar que Texas, Nueva York y Pensilvania son los estados con el mayor número de transacciones, tanto en número como en cantidad de dinero. Esto puede deberse a varios factores, como una alta densidad de población, un mayor número de comerciantes o un nivel de gasto per cápita más alto. Sin embargo, también notamos que estos tres estados son también los líderes en términos de transacciones fraudulentas, tanto en número como en cantidad de dinero. Esto sugiere que la frecuencia de las transacciones fraudulentas puede estar relacionada con la frecuencia total de las transacciones.

En cuanto a las transacciones fraudulentas, es interesante observar que, aunque California se encuentra en el cuarto lugar en términos de número de transacciones y cantidad de dinero, baja al séptimo lugar en el número de transacciones fraudulentas y al octavo en la cantidad de dinero en dichas transacciones. Esto puede indicar que California tiene mejores medidas de seguridad y prevención de fraudes en comparación con otros estados con un volumen de transacciones similar.

El análisis de la estructura y el contenido de los datos nos ha proporcionado una visión valiosa de las características y patrones de las transacciones, tanto generales como fraudulentas, y ha destacado varias áreas clave para tener luego en cuenta

## **Invocación de la API**

El objetivo primordial del proyecto consiste en diseñar un modelo que permitiera a los clientes detectar actividades fraudulentas de manera eficiente y efectiva. Para lograr esto, creamos una API que se invoca a través de una URL, la cual actúa como un disparador ('trigger') de nuestra función Lambda en AWS. Para probar y visualizar el funcionamiento de la API, utilizamos Postman, una herramienta ampliamente reconocida para el desarrollo y prueba de APIs.

El proceso de invocación de la API es simple: el usuario envía los datos de una transacción a través de Postman y, a cambio, recibe una respuesta que indica la probabilidad de que la transacción sea fraudulenta. Este resultado se basa en un análisis exhaustivo de datos y en un modelo de regresión logística entrenado previamente.

La regresión logística es una técnica de análisis de datos que se utiliza para predecir la probabilidad de ocurrencia cierto evento binario, en nuestro caso, si una transacción es fraudulenta o no (variable objetivo). Este método requiere una cuidadosa selección de variables independientes, que representan las características o atributos que se utilizan como entradas para el modelo y para predecir la variable objetivo. Teniendo en cuenta todas las columnas, seleccionamos aquellas que mejor se ajustan al modelo: categoría, amt (monto de la transacción), genero, zip, latitud, longitud. Aunque la hora también se consideró inicialmente, se decidió finalmente no incluirla, ya que su influencia en la precisión del modelo fue insignificante. Es importante mencionar que todas las variables de entrada deben ser numéricas. Por ello, transformamos las variables categóricas, como el género y la categoría, en números enteros. En particular, asignamos el valor 1 para 'masculino' y 0 para 'femenino'. Para la categoría, creamos un diccionario que asocia un número único a cada categoría existente, y mapeamos estos números a una nueva columna en nuestro conjunto de datos, llamada 'category\_num'.

Una vez preparado el conjunto de datos, se dividió en dos partes: el 70% de los datos se utilizó para entrenar el modelo, y el 30% restante se utilizó como conjunto de prueba para evaluar su rendimiento. Sin embargo, los resultados iniciales no fueron muy prometedores: aunque el modelo fue capaz de predecir correctamente la mayoría de las transacciones no fraudulentas, falló en la detección de la mayoría de las transacciones fraudulentas. Tras una revisión más detallada, evaluando diversas formas de mejora, descubrimos que solo el 0,39% de las transacciones en nuestro conjunto de datos eran fraudulentas, lo cual explicaba el rendimiento pobre del modelo en la detección de fraude. Es por ello que optamos por realizar un sobremuestreo sobre la clase minoritaria, en este caso aquellos que eran fraude.

Para llevar a cabo este sobremuestreo, decidimos aplicar una técnica de sobremuestreo conocida como SMOTE (Synthetic Minority Over-sampling Technique) para equilibrar las clases. SMOTE genera nuevas instancias sintéticas de la clase minoritaria, mejorando así la capacidad del modelo para aprender y predecir correctamente casos de fraude. Después de aplicar SMOTE, reentrenamos nuestro modelo con el nuevo conjunto de datos balanceado y volvimos a evaluar su rendimiento con el conjunto de prueba original. Como resultado, la precisión de nuestro modelo en la detección de transacciones fraudulentas mejoró significativamente.

Para la evaluación del modelo desempeñamos la matriz de confusión que nos brinda una representación visual de la precisión de las predicciones realizadas por el modelo al compararla con sus valores realmente. Consta de 4 secciones. Verdaderos positivos aquellos que el modelo predijo correctamente la presencia de una clase especifica. Verdaderos negativos donde se predijo correctamente la ausencia de una cierta clase. Luego, falso positivos y falso negativo donde se predijo de forma incorrecta la presencia o ausencia de una clase respectivamente. Una evaluación ideal sería el caso en que ambos verdaderos tengan los valores más altos. Observando ambas matrices de confusión (referencia 4.0 y 5.0) podemos apreciar lo que mencionábamos previamente donde el modelo inicial no logra clasificar correctamente ningún caso de fraude, lo cual es una limitación significativa. Mientras tanto, el segundo modelo se ajusta mejor y logra clasificar casos de forma correcta. Lo importante de la interpretación es que predijo correctamente que 157186 corresponden con no fraude y 502 con fraude.

El modelo de regresión logística fue realizado en la plataforma de Jupyter Notebook. Para llevarlo a AWS fue un gran desafío. Para ello utilizamos la librería Joblib, donde empaquetamos el modelo y lo subimos al Bucket de S3. En AWS, la función Lambda recupera el modelo del bucket de S3 y lo utiliza para hacer predicciones basadas en los datos enviados a través de Postman. Así, cada vez que se invoca la API, la función Lambda calcula y devuelve la probabilidad de que la transacción sea fraudulenta, proporcionando al usuario una valiosa herramienta para la detección de fraudes.

# **Lecciones aprendidas y posibles mejoras futuras**

A lo largo del desarrollo de este proyecto, nos enfrentamos a una serie de retos que nos permitieron adquirir nuevas perspectivas y aprendizajes en general. Además, identificamos oportunidades de mejora y eficiencia para aplicar en un futuro.

Respecto a la gestión de la base de datos, utilizamos una única tabla no normalizada con algunos datos redundantes, como, por ejemplo, los campos “unix time” y “trans\_date\_trans\_time”. En retrospectiva, esta redundancia podría haberse evitado. A su vez, existían datos que no empleamos en absoluto, como la ocupación del usuario, su fecha de nacimiento, y su nombre y apellido. Para optimizar este aspecto, una solución viable sería dividir la base de datos en dos tablas: una enfocada en los datos del usuario y otra en los datos de la transacción. Este enfoque nos permitiría concentrar nuestros análisis en los datos más relevantes.

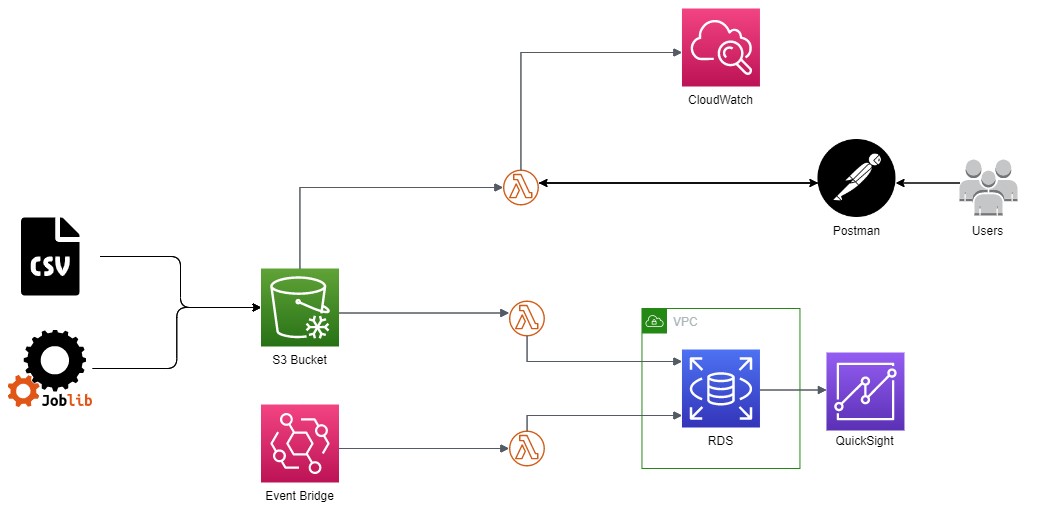
En cuanto a las bibliotecas utilizadas en las funciones Lambda, podríamos haber hecho un uso más eficiente de los recursos de AWS. En lugar de descargar estas bibliotecas en nuestros equipos personales y subirlas en formato ZIP, podríamos haberlas descargado directamente en una instancia EC2 existente y almacenado en un bucket de S3. Así, al crear la capa (layer) correspondiente, podríamos simplemente referenciar a ese bucket, optimizando así el tiempo y la memoria de nuestros recursos. Una alternativa adicional a este problema sería investigar acerca de las capas de código abierto (open-source) ya existentes. Esta estrategia también podría ayudar a reducir costos, ya que tanto la utilización de instancias EC2 como el almacenamiento en Bucket de S3 implican gastos que varían en función del tiempo de ejecución y el espacio de almacenamiento, respectivamente.

Además, para futuros proyectos, estaríamos interesados en explorar el uso de DynamoDB, la base de datos NoSQL de AWS. DynamoDB podría proporcionar una gestión de datos más eficiente y nos permitiría almacenar de manera sencilla y accesible ciertos tipos de transacciones de especial interés, como, por ejemplo, las transacciones fraudulentas de mayores cantidades de dinero.

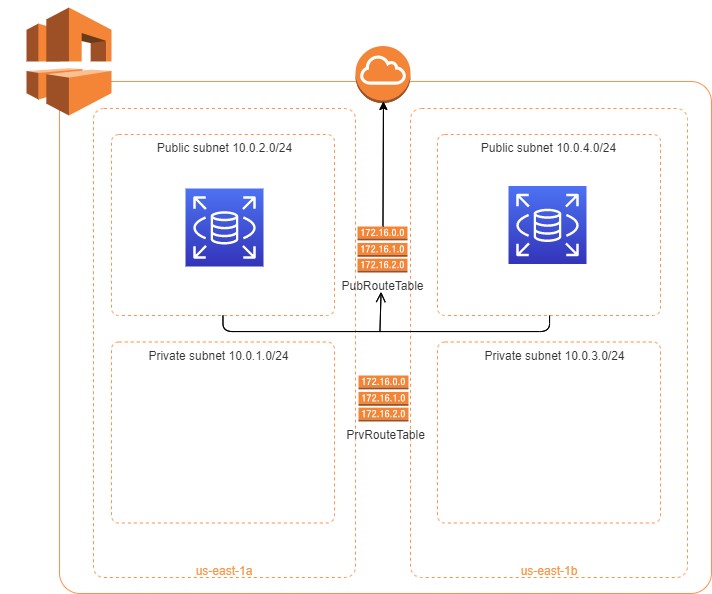
Por lo tanto, aunque llegamos a resultados valiosos y aprendimos mucho durante el desarrollo de este proyecto, también identificamos varias oportunidades para mejorar nuestros procesos y optimizar la utilización de recursos.

# **Anexo**

## **1.0 Diagrama de servicios utilizados**



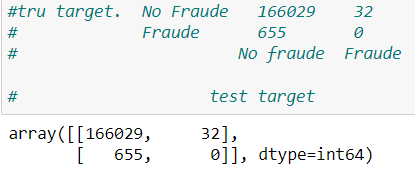
## **2.0 Diagrama de red**



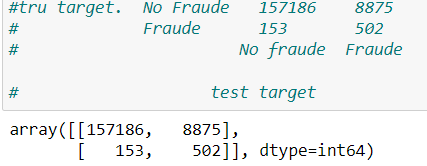
## **3.0 Estructura de la base de datos**



## **4.0 Matriz de confusión modelo inicial**



## **5.0 Matriz de confusión modelo final**



## **6.0 Repositorio de GitHub**

[Link al repositorio de GitHub con código relacionado al proyecto](https://github.com/agusBorges02/StacksBankFraud)