# Procesos de Machine Learning para la prevención de fraude en Mercadolibre

Autor: Gabriel Abdala
Tutor: Vanessa Welsh

Junio 2022, Ciudad Autónoma de Buenos Aires

# AGRADECIMIENTOS

En primer lugar quiero agradecer a Dios que me ha dado la vida y las diferentes bendiciones en mi vida.

A mi esposa Laura, mis hijos Matías y Josefina que siempre me apoyaron constantemente en los largos viajes de cursada, incluso en las horas de estudio en el hogar.

A mi padre que me mostró con su ejemplo el valor de estudiar y capacitarme. A mi madre que siempre me apoyó y animó para culminar esta etapa.

A Vanessa Welsh por guiarme en todo el proceso metodológico del presente trabajo.

A Mercado libre, por brindarme la posibilidad de realizar este MBA, apoyándome en tiempo y una beca económica.

A las personas entrevistadas de Mercado Libre para realizar el estudio de campo.

# RESUMEN

El *e-commerce* ha experimentado un gran crecimiento en los últimos años y se espera que siga incrementando. Muchos de los cambios de hábitos de consumo han cambiado potenciando el e-commerce. Gran cantidad de nuevos usuarios se han volcado a *e-commerce* debido a la pandemia y las cuarentenas utilizadas como mecanismos para evitar la propagación de COVID-19.

Muchos aspectos son facilitados para los usuarios por medio del *e-commerce* pero a su vez ha proliferado notablemente el fraude *online*, siendo perjudicados los usuarios y también las empresas.

La forma en la que grandes empresas de *e-commerce* atacan este problema es por medio del análisis de las transacciones con modelos de *machine learning*. Estos buscan predecir si la transacción es fraudulenta o no. Debido al notable incremento de fraude online es importante que el proceso de creación de estos modelos sea rápido, permitiendo así iterar frecuentemente para que los modelos detecten las nuevas modalidades de fraude.

El objetivo del presente trabajo es describir el proceso de *machine learning* utilizado por Mercadolibre, como mecanismo para la prevención de fraude y también hacer recomendaciones sobre dicho proceso.

Por medio de esta tesis se buscó dar respuesta a las siguiente preguntas:

- ¿Cómo se puede mejorar el proceso de *machine learning* en prevención de fraude, de manera tal que permita a estos procesos lograr mejores resultados en Mercadolibre?
- ¿Cómo es el proceso de *machine learning* en prevención de fraude en Mercadolibre?
- ¿Qué desafíos presentan estos procesos?

El análisis se realizó por medio de entrevistas a personas claves de equipos de prevención de fraude y a expertos del área.

Las principales conclusiones fueron las siguientes recomendaciones: Abordar un enfoque de autoservicio por medio de la creación/modificación herramientas de software, acelerar la adopción de un *feature store* y la implementación de un *data mesh*. Estas recomendaciones apuntan a mitigar o eliminar los principales desafíos o dolores.

**PALABRAS CLAVE:** Mercadolibre, machine learning, prevención de fraude**.**

# ÍNDICE GENERAL

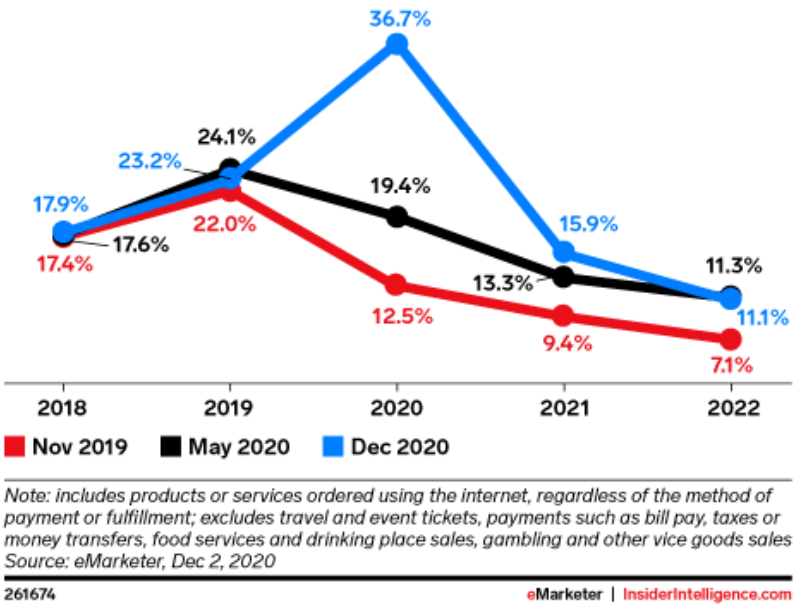
# ÍNDICE DE ILUSTRACIÓNES Y TABLAS

# INTRODUCCIÓN

Actualmente las ventas de *e-commerce* y *fintech* vienen aumentando en toda América Latina y se espera que para 2021 el comercio electrónico tenga tasas de crecimiento cercanas al 15.9% para Latam como es mostrado en la ilustración 1 (Ceurvels, 2020).

La ilustración Nº 1 muestra que en las diferentes predicciones de Nov2019, May2020 y Dec2020 las perspectivas de crecimiento fueron mejorando. La pandemia provocada por Covid-19 ha acelerado el crecimiento del *e-commerce* debido a la cuarentena o aislamiento que muchos países han establecido.

*Ilustración 1: Crecimiento esperado de ventas de e-commerce para Latinoamérica.*




*Fuente: (Ceurvels, 2020)*

Se estima que muchos de los cambios de hábitos del consumidor serán perdurables. Según Forbes (Enrico, 2020, párrafo 14) “Aunque hay muchas cosas que todavía no podemos anticipar, no quedan dudas de que el e-commerce saldrá fortalecido de esta crisis: las personas cambiarán sus hábitos de consumo y las empresas pondrán en valor la fidelidad y los gustos de sus clientes”.

El ecommerce brinda muchas oportunidades pero también facilita la proliferación de varios tipos de fraude, tanto a las empresas como a los usuarios de las mismas. Durante la pandemia se ha visto un notable incremento del fraude *online* de forma global (Radoini, 2020). Muchos países optaron por la cuarentena como mecanismo que frene la propagación de covid-19. Esto ha provocado que las personas pasen mucho más tiempo conectados a internet, como así también muchos nuevos usuarios para internet. Estos cambios han ocasionado una mayor actividad de crímenes electrónicos.

En un estudio realizado por Google y Atlas VPN (Google & Atlas VPN, 2020) uno de los resultados es un incremento de 350% de sitios dedicados a *phishing* desde Enero a Marzo (ilustración Nº 2). Esta es una forma común de fraude en internet que trata de engañar a los usuarios para que entreguen información personal relevante, un ejemplo de esto es contraseñas y tarjetas de crédito.

*Ilustración 2: Sitios de Phishing detectados por Google.*


*Fuente: (Google & Atlas VPN, 2020)*

En la ilustración Nº 2 se puede apreciar un incremento de un 96% en febrero versus enero y un 78% en marzo versus febrero. Esto hace un 350% de aumento de sitios dedicados a *phishing.*

En el reporte realizado por Merchants Savvy (Merchants Savvy, 2020) se observa que las pérdidas globales por pagos fraudulentos se han triplicado desde USD 9.84 billion

en 2011 hasta los USD 32.39 en 2020. Además se espera que este continúe creciendo hasta ser un 25% mayor en 2027 respecto a 2020, alcanzando los USD 40.62 billion.

Por lo antes mencionado se vuelve un factor clave para el éxito de cualquier compañía que entre al mundo *online* contar con una prevención de fraude adecuada. Mercadolibre es líder a nivel regional en *e-commerce* y un jugador muy importante en fintech. Desde, casi sus inicios, ha tenido un área encargada de brindar seguridad y confiabilidad a los usuarios de la plataforma como así también de proteger los intereses de la compañía.

Las personas que cometen fraude *online* están constantemente evolucionando en sus modalidades, las tecnologías que usan y la masividad de sus ataques. Es por ello también fundamental la mejora continua de los diferentes mecanismos de prevención de fraude. Mercadolibre arrancó en sus inicios con sistemas basados en reglas, diseñadas e implementadas por expertos y evolucionó a un sistema combinado por reglas y *machine learning*.

En el presente trabajo se buscó describir el proceso de machine learning en Mercadolibre, como mecanismo de prevención de fraude online como así también hacer propuestas de mejora.

Formulación del problema:

- ¿Cómo se puede mejorar el proceso de machine learning en prevención de fraude, de manera tal que permita a estos procesos lograr mejores resultados en Mercadolibre?

- ¿Cómo es el proceso de machine learning en prevención de fraude en Mercadolibre?
- ¿Qué desafíos presentan estos procesos?

La investigación fue realizada bajo un paradigma cualitativo, siendo la investigación de tipo descriptiva con un diseño metodológico no experimental, con estudio de caso único en profundidad de Mercadolibre.com. Los instrumentos de recolección de información

utilizados fueron entrevistas con personas claves de los equipos y expertos de la compañía.

El estudio se compone de tres capítulos:

En el capítulo uno se presentan las modalidades más comunes de fraude online y cómo la industria ataca esta problemática del ecommerce.

En el capítulo dos se describe el proceso de machine learning para prevención de fraude en Mercadolibre.

En el capítulo tres se presentan los resultados del estudio de campo realizado.

# CAPÍTULO I: TIPOS DE FRAUDE MÁS COMUNES EN *ECOMMERCE*

En el presente capítulo se explican las formas más comunes de cometer fraude en *ecommerce*. En los reportes de empresas líderes (LexisNexis, 2021; The Paypers, 2020; Columbus, 2020; Cybersource, 2019) del sector puede variar el orden de importancia que le asignan a cada tipo de fraude pero hay consenso sobre los más frecuentes.

## 1.1 Fraudes comunes en *ecommerce*

El fraude es un engaño a la víctima con la intención de sacar un beneficio económico. Los tipos de fraudes online más frecuentes son *Phishing, Account Takeover, Identity Thief* y *Credit Card Fraud.* A continuación se describe cada uno de ellos:

***Phishing:***

Esta modalidad es muy común (The Paypers, 2020) y sirve de base para otro tipo de ataques. Se denomina *phishing* al conjunto de técnicas que buscan engañar a sus víctimas haciéndose pasar por una persona o empresa de confianza, para obtener credenciales como usuario y contraseña, datos de tarjetas de crédito o información confidencial (*NIST*, 2007).

Las formas más usadas para atraer la atención de las víctimas es utilizar falsas promociones. En la Ilustración Nº 3 se muestra el mensaje de whatsapp que reciben las víctimas ofreciendo falsamente regalos con motivo del 20 aniversario de la compañía. Otra forma de atraer la atención de los usuarios es a través de aludir problemas del usuario con alguna empresa conocida que requiere su contacto.

*Ilustración 3: Mensaje de Whatsapp usado para phishing.*


*Fuente: (Infobae, 2021) estafa por whatsapp.*

En la Ilustración Nº 4 se muestra un sitio web usado para phishing utilizando como señuelo un problema en la cuenta del usuario. Un punto importante para destacar en esta figura es que la URL contiene una doble t ( "www.nettflix..." ) que no se percibe en una mirada rápida. El contacto inicial con la víctima puede realizarse de múltiples formas como emails (96%) , redes sociales/websites (3%) o sms/llamadas telefónicas (1%) o entre las más comunes (expertinsights, 2021).

*Ilustración 4: Sitio web que llama la atención del usuario por un problema en su cuenta.*


*Fuente: (Arteaga, 2017)*

**B. *Account Takeover*:**

Esta modalidad de fraude fue top 3 para ecommerce en 2019 (Cybersource, 2019). Este se produce cuando el atacante logra tener acceso a la cuenta de la víctima. Los fraudulentos obtienen información que les permite el acceso a cuentas de terceros por medio de *phishing*, a través de ataques de fuerza bruta, comprar bases de cuentas a *hackers, malware* (software malicioso que realiza acciones dañinas en un sistema informático sin el consentimiento del usuario), etc. Dependiendo del atacante en algunas ocasiones deja al dueño de la cuenta sin ningún tipo de acceso a la misma, cambiando los diferentes datos de acceso, en otras ocasiones prefieren pasar más desapercibidos y no realizar este tipo de acciones. Una vez que tome el control de la cuenta intentará monetizarla,

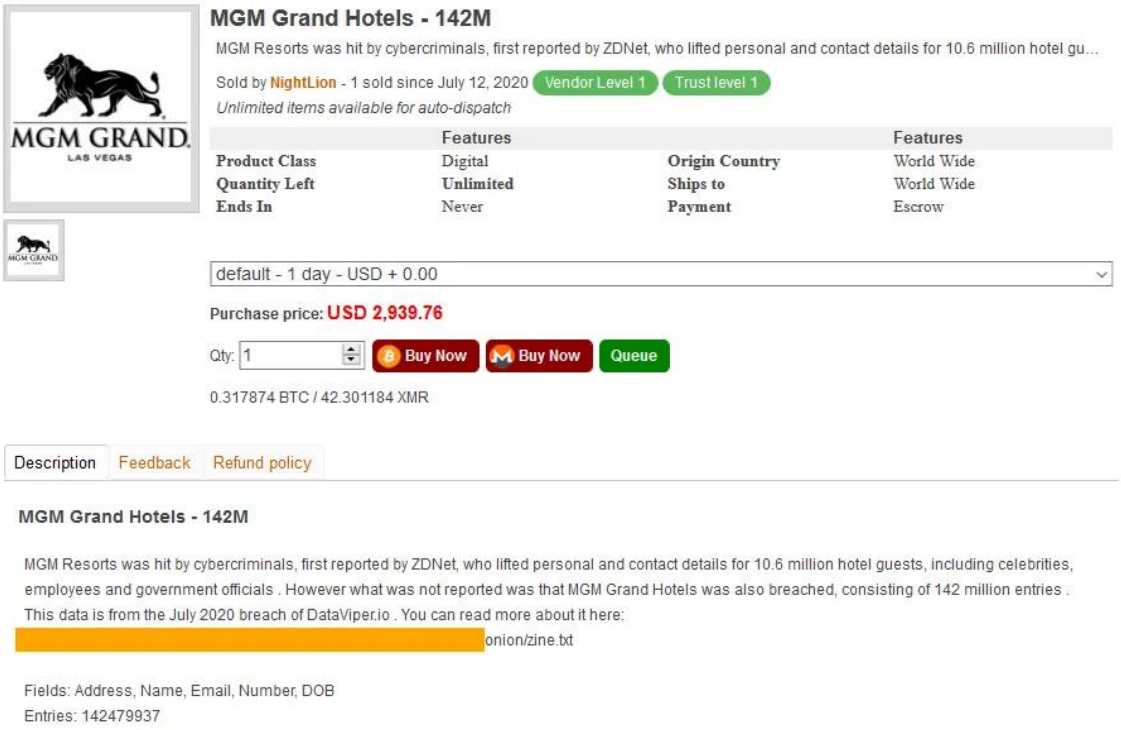
dependiendo del tipo de institución o empresa a la que pertenezca la cuenta variará la forma de hacerlo. En el caso de un banco el atacante intentará realizar transferencias a otras cuentas bancarias y otras acciones para vaciar de fondos la cuenta. Si fuera una cuenta de *ecommerce* además de sacar fondos (si tuviera), podría aprovechar la reputación de la cuenta para hacer ventas que nunca entregaría, pedir préstamos, hacer compras, etc.

### *C. Identity Thief*

Se llama así a la apropiación de la identidad de otra persona con el fin de obtener beneficios en su nombre. Este tipo de fraude ha sido una de los mayores vectores de ataque en 2020 (LexisNexis, 2021). Los datos personales puede obtenerlos el atacante por medio de phishing, una billetera o documentación robada en el mundo físico u online. Cuando el atacante posee una identidad robada podría crear cuentas en instituciones financieras para solicitar préstamos online, realizar compras en *ecommerce* con tarjetas robadas de la misma u otra persona, etc.

En la ilustración Nº 5 se puede observar la venta de la base de datos del hotel MGM Grand Las Vegas con 142 millones de registros de clientes por un poco más de usd 2900. Este tipo de *data breach* sirve como base para muchos fraudes de este tipo.

*Ilustración 5: Venta de datos en Dark Web*

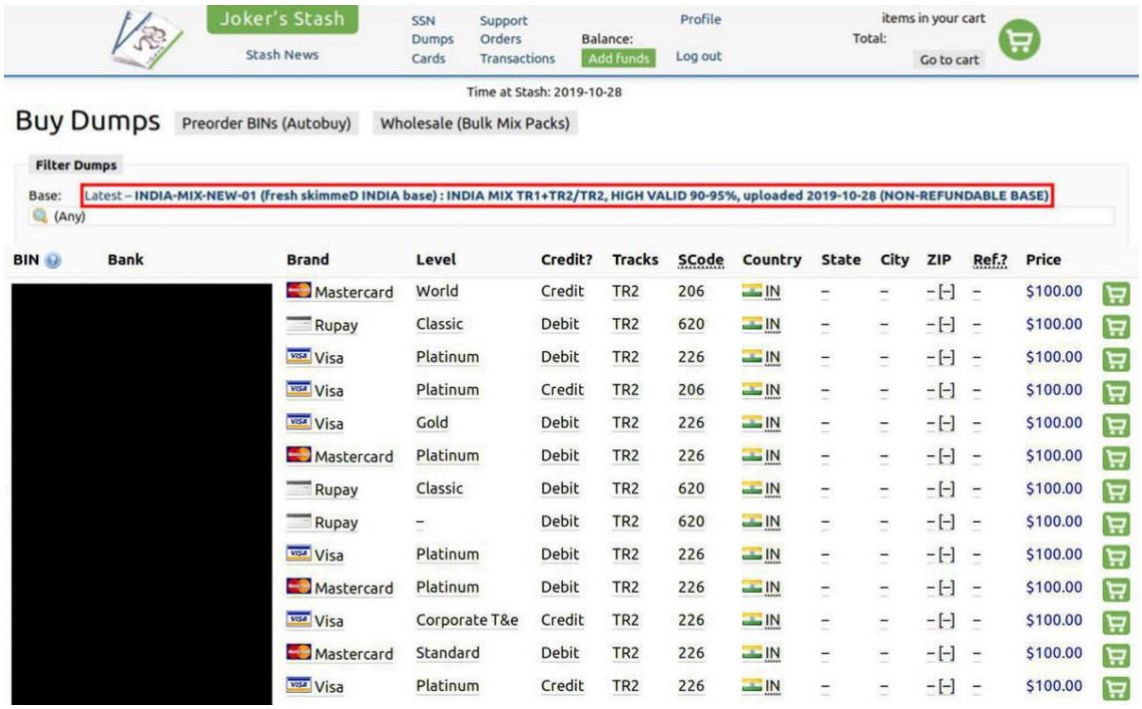



*Fuente: Catalin Cimpanu (Cimpanu, 2020)*

### *D. Credit Card Fraud*

Es la acción de realizar pagos con una tarjeta de crédito o débito robada con el fin de obtener bienes, servicios o enviar dinero a otra cuenta en posesión del atacante. Este tipo de fraude es el más común en *ecommerce* y generalmente el más costoso (Columbus, 2020). En este tipo de fraude, en *ecommerce*, puede ser perjudicado el dueño de la tarjeta, el sitio en el que se realiza la operación o la entidad emisora de la tarjeta. En el resumen de la tarjeta el titular puede visualizar pagos desconocidos por él y comunicarse con la entidad emisora para desconocerlos. Si dicha entidad entiende que el titular no efectuó el pago entonces el cargo se le anulará. Por un proceso de disputa establecido en la red de tarjetas de crédito se define si el costo de la operación lo termina pagando el ecommerce o la entidad emisora. Algunas formas en la que los atacantes obtienen tarjetas robadas puede ser por medio de *phishing* o comprar bases de tarjetas a otros *hackers* en *dark web* como es mostrado en la ilustración 6. En dicha ilustración se puede apreciar la venta de una base de tarjetas robadas en en *dark web*.

*Ilustración 6: Tarjetas de crédito robadas son vendidas en dark web.*




| BIN | Bank | Brand | Level | Credit? | Tracks | SCode | Country | State | City | ZIP | Ref.? | Price |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| [redacted] | | Mastercard | World | Credit | TR2 | 206 | IN | – | – | –[–] | – | $100.00 |
| | | Rupay | Classic | Debit | TR2 | 620 | IN | – | – | –[–] | – | $100.00 |
| | | Visa | Platinum | Debit | TR2 | 226 | IN | – | – | –[–] | – | $100.00 |
| | | Visa | Platinum | Credit | TR2 | 206 | IN | – | – | –[–] | – | $100.00 |
| | | Visa | Gold | Debit | TR2 | 226 | IN | – | – | –[–] | – | $100.00 |
| | | Mastercard | Platinum | Debit | TR2 | 226 | IN | – | – | –[–] | – | $100.00 |
| | | Rupay | Classic | Debit | TR2 | 620 | IN | – | – | –[–] | – | $100.00 |
| | | Rupay | – | Debit | TR2 | 620 | IN | – | – | –[–] | – | $100.00 |
| | | Visa | Platinum | Debit | TR2 | 226 | IN | – | – | –[–] | – | $100.00 |
| | | Mastercard | Platinum | Debit | TR2 | 226 | IN | – | – | –[–] | – | $100.00 |
| | | Visa | Corporate T&e | Credit | TR2 | 226 | IN | – | – | –[–] | – | $100.00 |
| | | Mastercard | Standard | Debit | TR2 | 226 | IN | – | – | –[–] | – | $100.00 |
| | | Visa | Platinum | Credit | TR2 | 226 | IN | – | – | –[–] | – | $100.00 |

*Fuente: (www.zdnet.com, 2019).*

## 1.2 Cómo ataca el ecommerce el fraude

El gran desafío que tiene la prevención de fraude en *ecommerce* es diferenciar las transacciones provenientes de usuarios genuinos de las que son originadas por fraudulentos. Se busca bloquear las transacciones fraudulentas sin agregar fricciones a los buenos usuarios. Típicamente el 99% de las transacciones son buenas pero el 1% de malas transacciones pueden ser muy costosas y hasta crecer si no se las acciona correctamente.

Los primeros sistemas de prevención de fraude consistían en un conjunto de reglas armadas por expertos que bloqueaban las transacciones de riesgo. Este tipo de enfoque tiene algunos problemas como dar muchos falsos positivos, es decir, bloquear muchas transacciones genuinas. Otro de los inconvenientes es el cambio de comportamiento de los usuarios, dejando obsoletos los umbrales bajo los cuales se activa la regla, esto

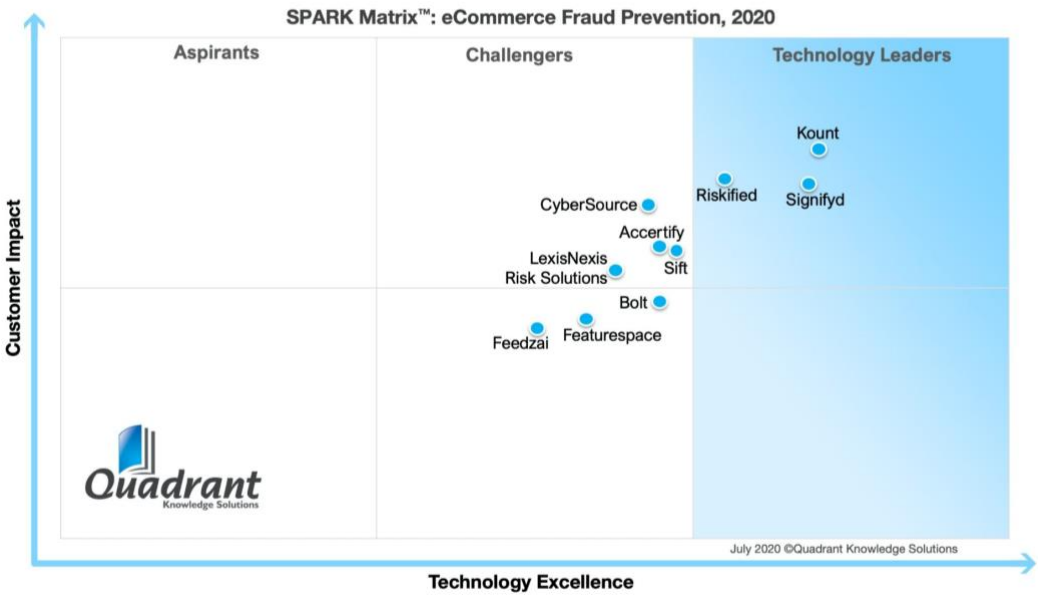
hace que se requieran cambios regularmente. Por último, es difícil de escalar a medida que se incrementan las reglas por los distintos patrones de fraude y sus mutaciones.

Las herramientas de prevención de fraude modernas combinan las reglas que puede hacer un experto con algoritmos de *machine learning*. Este último tiene la capacidad de aprender de los patrones en los datos y por ello tiene mucha más precisión que las reglas. También nos permite escalar ya que para *machine learning* cuantos más datos tengamos será mejor. La eficiencia es otra de las mejoras que vienen con ML ya que tiene mayor poder de análisis que un equipo de expertos y con costos mucho menores.

En la sección “Technologies and Innovations that keeps fraudsters at bay” (The Paypers, 2020, pag 15) del Fraud Prevention in Ecommerce Report se mencionan diversos proveedores líderes en soluciones de prevención de fraude. Estos comentan sus herramientas para atacar esta problemática y el factor común en ellas es el uso de ML.

En la ilustración Nº 7 se muestra el resultado del estudio de la industria de prevención de fraude realizado por Quadrant Knowledge Solutions (businesswire, 2020). En el eje X se presenta la excelencia tecnología y en eje Y el impacto en el cliente. Acorde a su nivel de excelencia tecnológica se cataloga a las empresas en aspirantes, retadores o líderes tecnológicos. Kount, es reconocido como #1 debido a que es la empresa con mayor excelencia tecnológica y el mayor impacto en el cliente. Uno de los principales factores que lo diferencia del resto es su avanzada inteligencia artificial con *machine learning*.

*Ilustración 7: Matriz resultante del informe de Quadrant Knowledge Solutions.*




*Fuente: (businesswire, 2020)*

Por lo tanto, de estos informes se puede concluir que la manera de atacar la problemática de fraude en *ecommerce* es utilizando *machine learning*. En el próximo capítulo se muestra el proceso estándar utilizado para generar modelos de *machine learning*.

# CAPÍTULO II: EL PROCESO DE MACHINE LEARNING

*Machine Learning* tiene muchos usos en la vida cotidiana, por ejemplo en la detección de email que son spam, recomendaciones de música, video o productos, reconocimiento de imágenes, etc. En este capítulo se explica qué es *machine learning*, sus usos y cómo se puede utilizar en la prevención de fraude.

## 2.1 Conceptualización Machine learning

*Machine Learning* hace referencia a una de las subdisciplinas de la inteligencia artificial en la cual las máquinas usan técnicas de aprendizaje estadístico con el fin de encontrar de forma autónoma patrones en los datos (Universidad de Alcalá, 2018). En otras palabras, *machine learning* le da a las computadoras la capacidad de aprender patrones de los datos sin ser programadas para esos patrones.

*Machine learning* es un proceso que toma un conjunto de datos y aprende los patrones en el mismo para posteriormente hacer predicciones en nuevos datos (Google, 2021).

Los modelos de *machine learning* se pueden subdividir en 3 grandes categorías (APD, 2019): aprendizaje supervisado, no supervisado y por refuerzo. El punto clave para distinguirlos son los datos que se utilizan en cada uno para su entrenamiento.

**a) Aprendizaje Supervisado:**

En este tipo de modelos los datos están etiquetados con los resultados que se desea aprender. Por ejemplo, si lo que se desea es que el algoritmo aprenda de los patrones de un email spam, se debe proporcionar un conjunto de emails en donde cada uno esté etiquetado con spam o no spam. Una vez concluido el entrenamiento se puede utilizar el modelos para predecir si un nuevo email (no presente en el conjunto de entrenamiento) es spam o no.

Dentro del aprendizaje supervisado tenemos los modelos de clasificación que intentan predecir las etiquetas de clase categórica de nuevos datos en base a los patrones aprendidos con el conjunto de entrenamiento. Por ejemplo, spam o no spam, ¿hay un gato en la foto?

Cuando lo que intentamos predecir es un valor numérico continuo, como el precio de una tarifa de taxi para un trayecto dado, lo que se usa es un modelo de regresión

**b) Aprendizaje no Supervisado**

En este tipo de modelos el conjunto de entrenamiento carece de etiquetas previas. La diferencia de este tipo de modelos es que solo toman en cuenta los datos de entrada y no los de salida para su proceso de entrenamiento. Estos tipos de algoritmos buscan patrones en los datos que permitan encontrar agrupaciones en los mismos. Se le conoce como *clustering* o *agrupamiento* (Torralba, 2021).

**c) Aprendizaje por Refuerzo**

En esta clase de algoritmos el aprendizaje viene por la experiencia. Estos intentan automatizar un procesos de decisión. En caso de que el resultado sea apropiado el algoritmo aprende a repetir esa acción en el futuro pero en caso de un resultado no deseado se evitará tomar esa decisión en un futuro (Instituto de Ingeniería del Conocimiento, 2021). Un caso de uso típico para este tipo de algoritmos es los autos autónomos.

En la ilustración Nº 8 se muestran, de manera resumida, las principales característica de las tres categorías existentes de *machine learning*

*Ilustración 8: Cuadro comparativo entre categorías de machine learning*

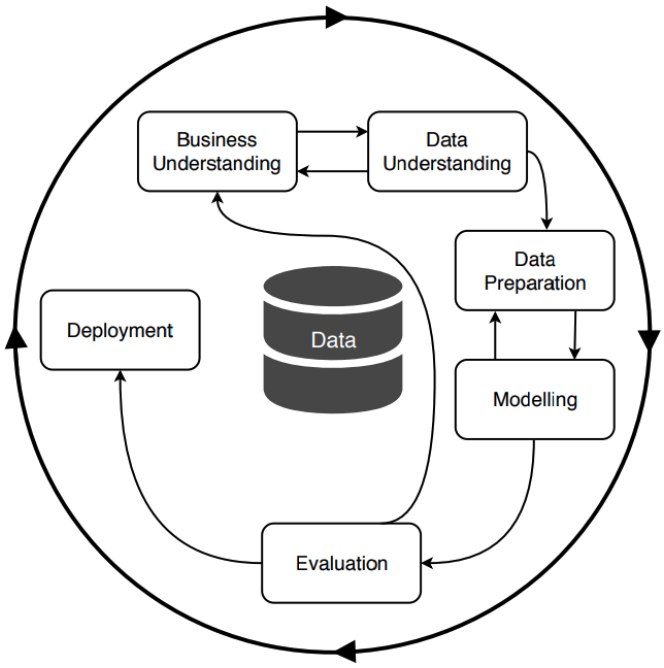
| **Categoría** | **Problema que resuelve** | **Inputs** | **Objetivo entrenamiento** | **Usos comunes** |
|---|---|---|---|---|
| Supervisado | Clasificación | Dataset etiquetado | Minimizar función de costo | Spam, Detección de Fraude |
| No Supervisado | Agrupamiento y asociación | Dataset sin etiquetas | Minimizar función de costo | Recomendaciones, Segmentación de clientes |
| Refuerzo | Aprender por experiencia | Datos como feedback | Maximizar beneficio | Autos Autónomos, Videojuegos |

*Fuente: Elaboración propia.*

## 2.2 Proceso estándar para la creación de Machine Learning

Hay un enfoque común sobre las diferentes partes que tiene el proceso para la creación de modelos de *machine learning*. El más conocido es Cross-Industry standard process for data mining CRISP-DM (Martínez-Plumed et al., 2019, 1). En la ilustración Nº 9 podemos ver las diferentes etapas del proceso.

*Ilustración 9: Etapas del proceso CRISP-DM*




*Fuente: (Martínez-Plumed et al., 2019, 2)*

A continuación se describen cada una de las etapas de **CRISP-DM** (Chapman et al., 2000, 13) presentadas en la ilustración Nº 8.

## 1. *Business Understanding*

En esta etapa se busca comprender los objetivos del proyecto y requerimientos desde el punto de vista del negocio. En base a esta comprensión se trata de plantear el problema desde la perspectiva de *machine learning* como así también hacer un plan inicial para alcanzar los objetivos planteados. Esta etapa se divide en 4 subetapas:

### 1.1. Determinar objetivos del negocio

En este paso se busca lo que el cliente quiere lograr. Se debe balancear entre objetivos y limitaciones. El analista debe encontrar los factores claves que pueden influir en el resultado del proyecto. Además del objetivo principal, frecuentemente existen otros temas relacionados al negocio que el cliente quiere abordar. Este paso es muy importante para no encontrar soluciones correctas para un problema diferente. Un aspecto clave de este paso es definir los *business success criteria* del proyecto desde el punto de vista del negocio.

### 1.2. Evaluar Situación

En esta parte se hace un análisis más detallado de lo que se vio de forma general en el punto anterior. Se indagará más detalladamente en los recursos (personas, software, etc), limitaciones (fechas, presupuesto, temas legales, etc), supuestos (sobre los datos y el negocio), riesgos y contingencias, costos y beneficios del proyecto y todo lo que deba ser tenido en cuenta para fijar correctamente el objetivo de *machine learning* y el plan del proyecto.

### 1.3. Definir el objetivo de *Machine Learning*

Se debe plasmar el objetivo de *machine learning* que nos llevará a cumplir con el objetivo de negocio. Se define en términos técnicos de *machine learning* el *Machine Learning Success Criteria*. El objetivo de

negocio puede ser “aumentar el profit del negocio” y el objetivo de *machine learning* “predecir transacciones fraudulentas” y así evitar pérdidas por fraude.

### 1.4. Plan del Proyecto

Se especifica un plan detallado del proyecto con sus diferentes etapas para lograr los objetivos de *machine learning* que nos permitirán alcanzar con los objetivo de negocio. También se plasmarán las herramientas y técnicas a utilizar.

## 2. *Data Understanding*

En esta etapa se hace una compilación de los datos para familiarizarse con ellos, pudiendo encontrar problemas de calidad en los datos, ganar los conocimientos iniciales sobre los datos o comenzar a formar hipótesis sobre información oculta en ellos. Esta etapa se divide en 4 subetapas:

### 2.1. *Collect Initial Data*

Se toman los datos listados en los recursos del proyecto y si fuera necesario se carga en la herramienta que se utilizará para su posterior entendimiento. Esta es una primera etapa de la preparación de datos, en la cual posiblemente tengamos que hacer adaptaciones para esta carga inicial.

### 2.2. *Describe Data*

Describir los datos especificando su formato, su cantidad. Un ejemplo de esto es la cantidad de registros y columnas de una tabla, los nombres de los campos u otra información característica de los datos. En este momento hay que indagar si los datos recolectados satisfacen los requerimientos más relevantes del proyecto.

### 2.3. *Explore Data*

En esta fase se exploran los datos por medio de consultas, visualizaciones y reportes. Se puede ver la distribución de atributos claves, relaciones entre los datos, probar agregaciones simples,

encontrar subpoblaciones importantes y análisis estadístico. Esto ayuda a entender la historia que están contando los datos. Se podrá entender si los datos son suficientes y relevantes para hacer un modelo.

### 2.4. *Verify Data Quality*

Se examina la calidad de los datos. Son de utilidad preguntas como: ¿Están los datos completos para cubrir todos los casos necesarios? En caso de contener errores en los datos ¿Cuán frecuentes son?. ¿Hay valores faltantes en los datos ? si fuera el caso, ¿cómo se representan? ¿Dónde ocurren y qué tan frecuentes son?. Las soluciones en la calidad de los datos generalmente depende en una buena parte de los datos y el conocimiento del negocio.

## 3. *Data Preparation*

En esta etapa están todas las tareas necesarias para construir el conjunto de datos (*DataSet*) a utilizar en el modelado. Las actividades de preparación de datos es probable que se realicen más de una vez y sin un orden prescrito. Las tareas incluyen: selección de tablas, registros y atributos, limpieza y transformación de datos. Esta etapa se divide en 5 subetapas:

### 3.1. *Select Data*

Definir los datos que se usarán para modelado. Es importante para esta selección ver la relevancia de los datos para el objetivo de machine learning, calidad y restricciones técnicas (volumen de datos y tipo). Se debe fijar la cantidad de atributos y registros a utilizar.

### 3.2. *Clean Data*

Se debe mejorar la calidad de los datos para que estén acordes a las técnicas de análisis seleccionadas. Esto podría implicar elección de subconjuntos de datos limpios, inserción de valores por defecto o los resultantes de un modelos de *machine learning* que prediga el valor faltante aprendiendo de los datos.

### 3.3. *Construct Data*

Esta tarea incluye la construcción de atributos derivados desde otros, incluir registros nuevos o transformados desde atributos existentes.

### 3.4. *Integrate Data*

Se busca combinar múltiples fuentes de información para crear nuevos registros o valores. Este paso también incluye las agregaciones sobre múltiples registros o fuentes de datos para sumarizar nuevos valores.

### 3.5. *Format Data*

Refiere a las transformación que se hacen en los datos (sintácticas y no de significado) para poder usar ciertas herramientas de modelado.

## 4. *Modeling*

En esta etapa varias técnicas de modelado son probadas y sus parámetros calibrados para encontrar su óptimo. Puede haber muchas técnicas para un mismo tipo de problema. Es posible que alguna técnica necesite los datos en un formato particular y esto haga que se tenga que volver a la etapa de preparación de datos. Esta etapa se divide en 4 subetapas:

### 4.1. *Select Modeling Technique*

Seleccionar la técnica de modelado a usar, se podría por ejemplo optar por un árbol de decisión, una red neuronal o un *random forest*. Dependiendo de la técnica que se seleccione puede que se necesite volver a la preparación de datos, ejemplo, las redes neuronales funcionan mejor con cierto formato en los datos que no afectan del mismo modo a los árboles de decisión.

### 4.2. *Generate Test Design*

En esta fase se generará un mecanismo para verificar la calidad y validez del modelo. Se deberá hacer un plan para entrenar, testear y validar el modelo. La primera parte del plan es como dividir el dataset para suplir a los 3 pasos. También es definir cómo mediremos la calidad del modelo, por ejemplo, en un modelo de clasificación suele usarse comúnmente error rates como medida de calidad.

### 4.3. *Build Model*

Construir el o los modelos con el dataset de entrenamiento y las herramientas de modelado seleccionadas. Dependiendo de la técnica elegida se deberá modificar diferentes parámetros para buscar el modelo óptimo.

### 4.4. *Assess Model*

Se evalúa el modelo contra el *machine learning success criteria* y el *dataset* de prueba y validación. En esta tarea se evalúan y comparan todos los modelos generados rankeandolos en base a *success criteria*. También se revisan los parámetros y se ajustan para una siguiente iteración de *build model* para así encontrar el mejor modelo posible.

## 5. *Evaluation*

En esta etapa ya se cuenta con uno o varios modelos con una buena calidad desde el punto de vista de análisis de datos pero es importante revisar los pasos seguidos para construir el modelo y así asegurarnos que pueda alcanzar los objetivos de negocio. Un objetivo clave de esta etapa es que no haya ningún asunto de negocio importante sin considerar. Esta etapa se divide en 3 subetapas:

### 5.1. *Evaluate Results*

En este paso se evalúa como el modelo cumple el *business success criteria* y se analiza si hay alguna razón por la cual el modelo es deficiente. Si el presupuesto y/o tiempo del proyecto lo permitiera se podría testear el/los modelo/s en una aplicación real.

### 5.2. *Review Process*

En este punto se hace un análisis completo del proceso de *machine learning* seguido para determinar si todos los pasos fueron ejecutados correctamente o algo se pasó por alto. En esta revisión se evalúan cosas como calidad de los datos, selección y construcción de los atributos, etc.

### 5.3. *Determine Next Steps*

Dependiendo de las etapas anteriores se debe decidir si se avanza a una etapa de despliegue, se hacen más iteraciones a los modelos o se establece un nuevo proyecto de *machine learning*.

## 6. *Deployment*

La creación del modelo no es el final del proyecto. A menudo implica poner el modelo dentro de procesos de toma de decisiones ya sea en tiempo real o de forma *batch*. Dependiendo de los requisitos del proyecto la fase de implementación puede ser simple o tan compleja como un proceso de *machine learning* repetible para toda la compañía. Esta etapa se divide en 4 subetapas:

### 6.1. *Deployment Plan*

El objetivo de este punto es crear una estrategia de despliegue del modelo en el entorno real. El plan contiene la estrategia, los pasos necesarios y cómo ejecutarlos para el correcto despliegue. En caso de haber un procedimiento para crear el modelo, es en esta etapa que se documenta para posteriores despliegues.

### 6.2. *Monitoring and Maintenance Plan*

Esta etapa es muy importante si los resultados del modelo son una parte fundamental del funcionamiento del negocio. El correcto mantenimiento y monitoreo evita que pasen largos periodos de tiempo con problemas en el entorno productivo que impacten en los resultados de negocio. En este punto se arma el plan del proceso de monitoreo y mantenimiento con su estrategia y pasos a seguir.

### 6.3. *Produce Final Report*

Al final del proyecto se prepara un reporte final incluyendo todos los entregables intermedios y un resumen de los resultados de *machine learning*.

### 6.4. *Review Proyect*

Se hace una revisión sobre que se hizo bien, que se hizo mal y que se puede mejorar en futuros proyectos.

El proceso descrito anteriormente hace uso de diferentes tecnologías para facilitar la creación de modelos de *machine learning*. Se mencionan algunas de ellas en la próxima sección.

## 2.3 Tecnologías usadas en Machine Learning

Los grandes de la industria del software utilizan 3 tecnologías en sus procesos de machine learning. Las mismas fueron seleccionadas por ser aceleradores de los procesos de machine learning una vez adoptadas. Estas tecnologías son de mucha utilidad en los modelos supervisados de machine learning utilizados para la prevención de fraudes.

### 2.3.1 Feature Store

Un *feature* es cualquier atributo medible que se pueda utilizar en un modelo predictivo. Un ejemplo de esto puede ser las transacciones del usuario en la última hora. *Feature store* es el lugar donde los features son guardados y organizados para ser usados en el entrenamiento de modelos y las posteriores predicciones de los mismos (FeatureStore.org, n.d.).

Actualmente los grandes proveedores de servicios en la nube proveen este tipo de soluciones entre los servicios que ofrecen para acelerar el proceso de *machine learning* de sus clientes. Google Cloud Platform (GCP) ofrece Vertex IA Feature Store, Amazon Web Services (AWS) ofrece SageMaker Feature Store y Microsoft Azure ofrece Azure Databricks Feature Store.

Varias empresas han visto mejoras en sus proceso de *machine learning* por la adopción de esta tecnología. Alguna de ellas son:

- Atlassian, pudo acelerar el tiempo para construir y poner el producción features de 1 - 3 meses a 1 día. Mejoró al precisión de predicción de sus modelos existentes en un 2%. (Tecton, 2021)

- iFood, que logro reducir un 50% su base de código utilizada para el cálculo de features histórico y online. Redujeron el número de procesos necesarios para estas tareas de docenas a 10. (Holds, 2020)

### 2.3.2 Data Mesh

Este nuevo enfoque sobre los datos busca minimizar o eliminar las falencias de los enfoques centralizados y monolíticos que se usan en las plataformas de datos actuales. Este concepto arrancó en 2019 por Zhamak Dehghani (Dehghani, 2019) cuando trabajaba como consultora principal en ThoughtWorks. Este es un enfoque relativamente nuevo en el mundo de las plataformas de datos pero que ya algunos grandes de la industria han implementado, entre los más conocidos están:

- Netflix (Tiagi et al., 2021), el equipo de trabajo comentó sobre la adopción de esta tecnología “Con la última plataforma de data-mesh, el movimiento de datos en Netflix Studio alcanzo una nueva etapa. Este plataforma guiada por la configuración reduce significativamente el lead time cuando se crea un nuevo pipeline, mientras ofrece nuevas características de soporte end-to-end a la evolución del esquema, interface de usuario self-service y acceso seguro a los datos”
- Roche (Dataops.live, s.f.), el Head of Data Platform de Roche Diagnostics Paul Rankin resume sobre esta implementación de data-mesh “Es un completo game changer. Hablamos de ROI en términos de miles de horas y dólares ahorrados”

**Data Mesh se basa en 4 principios:**

- **Descentralización de la propiedad de los datos y orientados al dominio:** esto significa que la propiedad de los datos será de los equipos del dominio que pueden ser departamentos o unidades de negocio. En lugar de tener un equipo centralizado que gestiona los datos, cada dominio deberá ser responsable por sus datos de punta a punta a fin de proveerlos como producto a la organización.
- **Datos como producto:** esto implica que los mismos sean:
    - Detectables, los dominios de datos deben ser fácilmente encontrados por los que los necesiten.
    - Direccionables, deben poder ser accedidos de forma programática o automática para su uso.

  - Confiables, deben tener calidad los datos.
  - Auto Descriptivos, contar con la documentación adecuada sobre la semántica de cada dato, sus tipos, dueños y todo lo que ayuda a su mejor uso.
  - Interoperables, se debe poder distribuir la información y unirla a otros dominios o enriquecerse para potenciar los análisis.
  - Seguros, proveer de controles de acceso de forma automática.
- **Plataforma de datos como autoservicio:** provee los servicios para que los dueños de los dominios puedan velar por los datos de punta a punta, garantizando la calidad y disponibilidad a los consumidores como un producto.
- **Gobernanza computacional federada:** se busca equilibrar el control centralizado pero permitir que la toma de decisiones esté lo más cerca del dominio como sea posible. Se busca que la codificación y la ejecución de políticas para todos y cada uno de los datos como productos sean automatizadas.

### 2.3.3 AutoML

AutoML (Saurav Singla, 2020) es una forma de automatizar algunas etapas del proceso de ML, entre las más beneficiadas están la elección del tipo de modelo a usar, su arquitectura interna y el ajuste de hiperparámetros.

En el presente capítulo se presentó el proceso estándar utilizado para la creación de modelos de *machine learning*. En el próximo capítulo se presentan los resultados del estudio de campo, en el cual se busca entender el proceso de *machine learning* utilizado por Mercadolibre para la creación de sus modelos. Así también, encontrar oportunidades o dolores en este proceso y finalmente se presentan recomendaciones para mitigarlos.

# METODOLOGÍA DE INVESTIGACIÓN

La investigación se realizó bajo un paradigma cualitativo. Siendo la investigación de tipo descriptiva con un diseño metodológico no experimental. Para la recolección de información se utilizaron entrevistas a personas claves y expertos de Mercadolibre, el cual es el objeto de estudio.

En la ilustración N° 10 se listan las personas claves que fueron entrevistadas para relevar el proceso de *machine learning* de prevención de fraude de Mercadolibre.

*Ilustración 10: Personas claves entrevistadas*

| **Nombre** | **Cargo** | **Equipo** | **Fecha Entrevista** |
|---|---|---|---|
| Persona clave 1 | IT Manager | Marketplace Payments | 9/9 - 28/9 |
| Persona Clave 2 | IT Project Leader | Online Payments | 16/9 - 20/9 |
| Persona Clave 3 | IT Project Leader | Wallet Payments | 2/9 - 21/9 |

*Fuente: Elaboración propia*

En todas las entrevistas a personas claves de equipos que usan machine learning para la prevención de fraude en la fintech de Mercadolibre las preguntas fueron las especificadas en el anexo 1.

Asimismo se procedió a entrevistar a expertos de *machine learning* dentro de Mercadolibre para encontrar propuestas de solución a cada uno de los mayores dolores que experimentan los equipos. En la ilustración N° 11 se especifican los expertos entrevistados.

*Ilustración 11: Expertos entrevistados*

| **Nombre** | **Cargo** | **Equipo** | **Fecha Entrevista** |
|---|---|---|---|
| Experto 1 | Machine Learning Director | Fraude Fintech | 13/04/2022 |
| Experto 2 | Machine Learning Sr Expert | Machine Learning Platform | 22/04/2022 |

*Fuente: Elaboración propia*

En el anexo 2 se presentan las preguntas realizadas a los expertos.

# CAPÍTULO III: PROCESO DE MACHINE LEARNING DE PREVENCIÓN DE FRAUDE EN MERCADOLIBRE

Mercadolibre es una empresa fundada en 1999 por Marcos Galperin y un grupo de emprendedores con el objetivo de revolucionar el comercio electrónico en América Latina. Con el paso de tiempo se fueron agregando nuevas unidades de negocio cómo:

- Mercadopago: Ofrece a compradores y vendedores múltiples servicios financieros como realizar o recibir pagos. Permite a los usuarios realizar o recibir pagos de modo online pero también en el mundo físico. Es una billetera virtual que brinda a sus clientes, entre otras características, la posibilidad de tener su dinero invertido generando ganancias a la vez de tenerlo disponible para utilizarlo.
- Mercado Ads: pensado para que los mejores vendedores del ecosistema mejoren la visibilidad de sus productos.
- Mercado Envíos: permite a los usuarios de mercadolibre optimizar las entregas a compradores y vendedores. Al usar Mercado Envios los paquetes cuentan con una total protección para ambos lados de la transacción.
- Mercado Crédito: Brinda créditos a los compradores y vendedores para ser usados en la plataforma o fuera de ella.
- Seguros: Solución de seguros que permite extender garantía a productos comprados en Mercadolibre. También se ofrecen seguros ante robo o daño en algunos productos.

Todas estas unidades se apalancan entre sí brindando mucho valor al usuario y fortaleciendo el ecosistema.

Los diferentes productos digitales que Mercadolibre provee a los usuarios tienen que estar protegidos del creciente fraude online para asegurar una buena experiencia a sus clientes como también así evitar pérdidas financieras a la compañía.

Como se mencionó en el capítulo 1.2 la herramienta diferencial a la hora de atacar el fraude online es el uso de *machine learning* para diferenciar las transacciones buenas de las fraudulentas.

## 3.1 *Machine Learning y su uso en prevención de fraude*

De acuerdo a las entrevistas a las personas claves se pudo saber que los algoritmos de *machine learning* son capaces de aprender los comportamientos correctos e incorrectos, posteriormente estos modelos se usan para predecir el riesgo de una transacción en base a lo aprendido de transacciones previas. Estos modelos pueden detectar comportamientos sospechosos incluso en usuarios que nunca usaron la plataforma ya que aprenden los patrones de comportamiento fraudulentos.

Uno de los casos de uso de Mercadolibre es la detección de fraude, para ello se le provee a los algoritmos de un conjunto de datos con transacciones genuinas y fraudulentas para que pueda encontrar los patrones de los casos no deseados. Una vez que el modelo de *machine learning* ha aprendido de los datos está listo para hacer predicciones en futuras transacciones.

Los fraudes tienen un alto impacto económico en las finanzas de Mercadolibre. Las provisiones de chargebacks (pagos con tarjeta de crédito no reconocidos por el titular) al 31 de diciembre fueron de USD 11.3M, USD 17,6M y USD 13.9M para el 2019, 2020 y 2021 respectivamente. (Mercadolibre.com, 2022)

Los atacantes están continuamente cambiando y encontrando nuevas formas de hacer fraude por lo cual se vuelve muy importante hacer que los algoritmos de machine learning aprendan rápidamente estos nuevos patrones. Es importante reentrenar/crear modelos en menos tiempo.

En el presente capítulo se describe el proceso que usa Mercadolibre para hacer machine learning dentro de prevención de fraude en pagos. También se sugieren mejoras al proceso.

## 3.2 Proceso de Machine Learning en prevención de fraude en Mercadolibre

El proceso que sigue prevención de fraude en Mercadolibre es el descrito por **CRISP-DM** (Chapman et al., 2000, 13) en el capítulo 2.2 de la presente tesis . Para describir

cada una de las etapas se entrevistó a personas claves de los equipos de prevención de fraude. A continuación se explican cómo se realizan estas etapas:

### 3.2.1 *Business Understanding*

De acuerdo a los resultados de las entrevistas, surgió que para la definición de los objetivos del proyecto se comienza con las reuniones con los sponsors de negocio para conocer el producto, sus características y los riesgos potenciales visualizados por el sponsor. Posteriormente se hace un reunión de *Threat Modeling* para analizar los riesgos antes mencionados más lo que puedan encontrar los expertos de fraude. En este punto se analiza junto con los expertos de qué forma machine learning puede resolver la problemática y aportar valor al usuario final. En estas reuniones se ve la viabilidad del proyecto contemplando riesgos, costos y beneficios, también se acuerda el *business success criteria* y las fechas de entrega acordadas en caso de avanzar con el proyecto.

El *business success criteria* generalmente está asociado al porcentaje de aprobación de transacciones y el ratio de fraude. El proyecto debe buscar maximizar el primero y minimizar el último. También puede que estos criterios de éxito se enfoquen en algún segmento específico como usuarios nuevos o alguna categoría de productos determinada.

El *machine learning success criteria* es definido por el equipo de fraude acorde al entendimiento del producto y de la etapa en la que se encuentra. Las métricas a considerar para este criterio serán *Accuracy, Recall, Confusión Matrix, AUC y F1- Score*.

El tiempo y complejidad de esta etapa puede variar dependiendo de si es un proyecto nuevo o un reentrenamiento de un modelo existente. En el primer caso generalmente se va haciendo el análisis a la par de la definición y construcción del producto, pudiendo durar desde 1 a 3 meses, sin ser este un tiempo continuo de entendimiento. En el caso de un nuevo entrenamiento de un modelo existente puede definirse en un par de reuniones con los sponsors de negocio.

### 3.2.2 *Data Understanding*

En esta etapa se busca entender qué fuentes de información existen y pueden ser usadas para resolver el problema planteado. Esta información puede estar en bases de datos relacionales, NoSql como datos no estructurados, en archivos de logs en el *lake* de datos o en el *warehouse*. Un punto importante a tener en cuenta es verificar que la información a utilizar cumpla con normas de PII (información de identificación personal) vigentes. Este último aspecto puede llevar a desestimar una fuente de datos o tener que hacer adaptaciones para poder utilizarla.

Una vez que se cuenta con las diferentes fuentes de datos a utilizar se describe a cada una de ellas especificando tecnología, estructura e información característica de los datos. También se indaga cómo unir la información proveniente de diferentes fuentes.

Para una primera exploración de los datos generalmente se utilizan consultas SQL y después se utiliza *jupiter notebooks* para visualizar los datos, hacer reportes y gráficos que permitan ir generando entendimiento de los datos y las primeras hipótesis sobre los mismos. En caso de explorar un nuevo dato a incorporar al modelado, la verificación de la calidad y consistencia del mismo se hace manualmente con el uso de SQL y dependiendo del criterio del científico de datos.

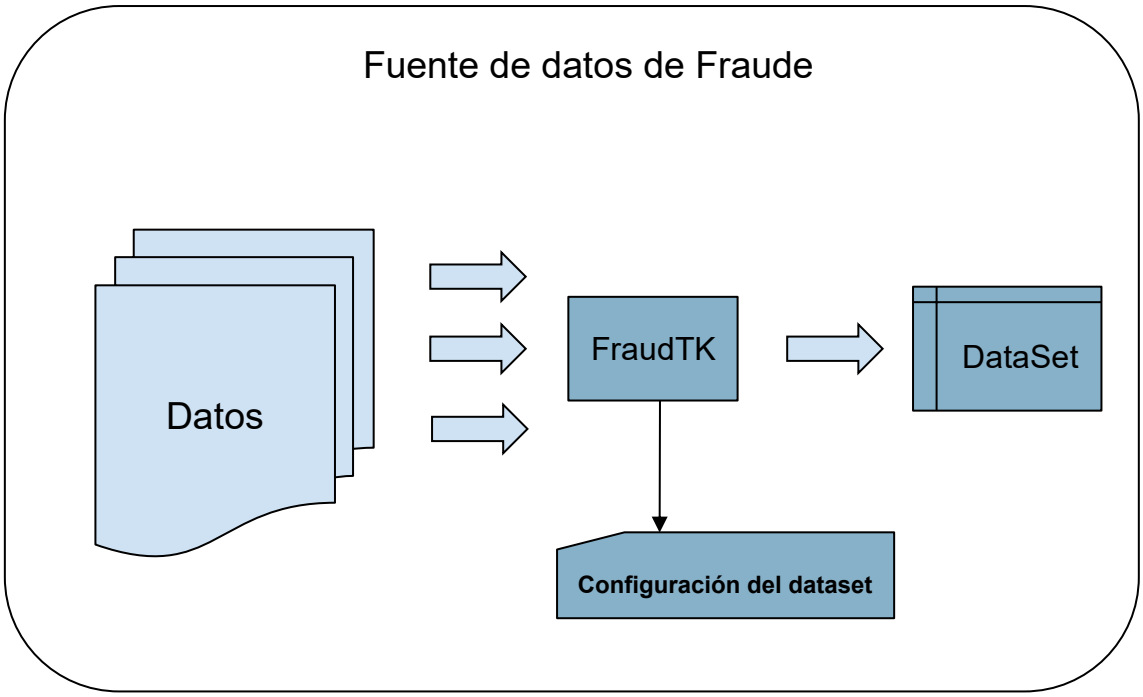
Uno de los principales problemas de esta etapa es la multiplicidad de fuentes y formatos de datos. Se está avanzando en la dirección de tener una gran fuente de datos como BigQuery pero no todos los departamentos vuelcan toda su información en BigQuery. Otro problema es la falta de documentación en la que se especifique cómo se debería usar un dato, cuál es la calidad y disponibilidad esperada en producción. Esta falta de documentación no solo se da en fuentes externas a fraude sino también en entidades construidas por diferentes equipos de fraude.

El tiempo insumido en esta etapa puede variar entre 1 y 3 semanas dependiendo del proyecto si es un proyecto nuevo, un reentrenamiento, la diversidad de fuentes que se necesite acceder y formato de los datos.

### 3.2.3 *Data Preparation*

En esta etapa se busca armar el *dataset* que se utilizará posteriormente en el modelado (Ilustración N° 12). Para ello se cuenta con la herramienta FraudTK. Esta librería utiliza una configuración de los datos a incluir para armar el *dataset*.

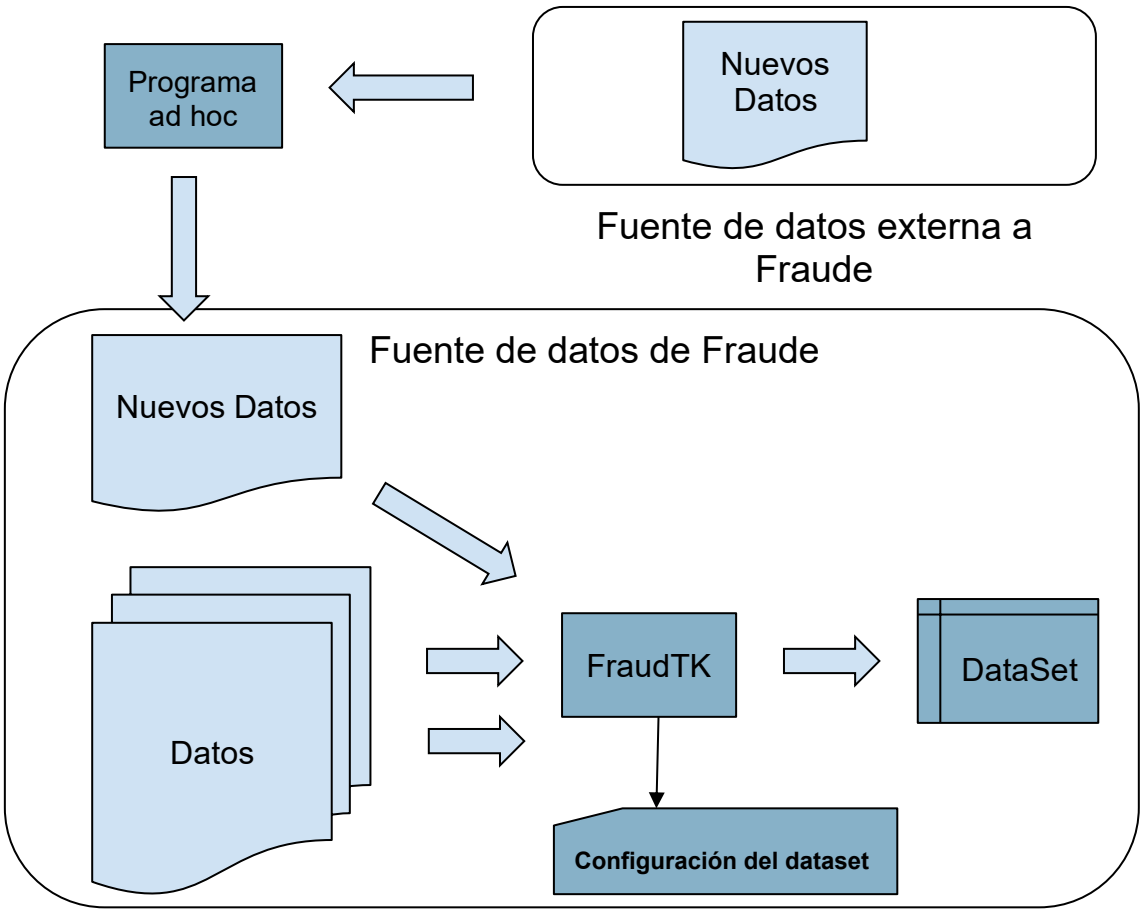
*Ilustración 12: Proceso de armado del dataset.*




*Fuente: Elaboración propia*

En la ilustración Nº 11 se muestra una simplificación del proceso de generación del *dataset*. En la fuente de datos de fraude se almacena información histórica de cada transacción para futuros entrenamientos de modelos. Esta cuenta con múltiples entidades que aportan a la prevención de fraude. Una entidad puede ser Pagos, Direcciones, Órdenes de compra, etc. En la configuración del *dataset* se especifican las entidades y campos a usar y también las transformaciones a realizar en los datos. Con esta configuración FraudTK realiza las uniones y transformación de los datos resultando en el *dataset* (Ilustración N° 13). Una vez terminado el proceso se muestran estadísticas que permiten verificar la calidad del *dataset* armado.

*Ilustración 13: Proceso de armado del dataset con fuentes externas.*




*Fuente: Elaboración propia*

El caso más simple es cuando se requiere entrenar un modelo con la información existente sin agregar nuevas entidades. En la ilustración Nº 12 se muestra el mismo proceso pero con la variante que se incorpora una o varias entidades desde una fuente externa a fraude. Para este escenario es necesario hacer un programa *ad hoc* que tome la información desde la fuente externa, realice transformaciones si fuera necesario y lo guarde en la fuente de datos de fraude. El siguiente paso será agregar una nueva configuración para que FraudTK pueda incluir esta nueva información en el armado del dataset.

FraudTk provee estadísticas que permiten verificar la calidad de los datos. Si la revisión de la calidad de los datos es satisfactoria se pasa a la etapa de modelado pero en caso de encontrar problemas puede llegar a tener que armar nuevamente el dataset. Esto ha acontecido algunas veces por inconsistencias en la información ya utilizada por el modelo o por el hecho de que los controles manuales de la calidad del nuevo dato no

fueron correctos o exhaustivos. Cuando esto ocurre hay que generar nuevamente el dataset, incurriendo en incremento de costos y tiempo.

Con una versión validada del dataset se realiza la selección de los mejores variables para predecir si una transacción es fraudulenta. Se analiza con diferentes algoritmos para dejar las variables más representativas. Una vez realizado este proceso se eliminan del dataset las variables que no ayudan a diferenciar buenos de malos usuarios. A esta parte se la conoce como *feature selection.*

La multiplicidad de fuentes de datos y tecnologías tiende a complicar y alargar el proceso. Estos programas *ad hoc* suelen hacerse en python o java, siendo esta parte del proceso muy artesanal y propensa a errores que llevan a retrabajo.

Dependiendo del país y producto, el volumen de datos del *dataset* puede ser significativo. En algunas oportunidades se da que hay muchos procesos armando *datasets* y ante la necesidad de recrearlo hay que esperar hasta el siguiente día.

La duración de esta etapa puede variar entre 2 a 7 días dependiendo si es un proyecto nuevo, uno existente que agrega nuevos datos o un reentrenamiento del modelo actual.

En esta etapa se construyen las variables nuevas que posteriormente estarán disponibles al momento de hacer predicciones en producción. Inicialmente se hace un análisis para ver si la nueva variable ayuda a diferenciar las transacciones fraudulentas de las que no lo son. En caso de tener un buen poder predictivo se agrega la configuración en FraudTK para que la tenga en cuenta al armar el *dataset*. Los ingenieros de datos desarrollan esa variable a nivel productivo. A fin de poder agregar esta variable al *dataset* a utilizar se realiza una sintetización de la misma, esto es una reconstrucción histórica para cada transacción en el *dataset*. Cuando está disponible en producción, se realiza una comparación con la sintetización que se realizó para el *dataset* a fin de verificar su correctitud y evitar retrabajos en la etapa de *modeling* o *deployment*. Uno de los problemas que presenta este punto es la gran variedad de tecnologías con las que se construyen estas variables y el alto acoplamiento de los modelos a los datos. Algunas variables están en tecnologías modernas y otras no tanto. Al momento de hacer un cambio tecnológico en una variable es probable que se necesite reentrenar el modelo si la distribución de la variable no fuera muy similar a la

versión anterior. Esto puede implicar un gran trabajo debido a la gran cantidad de modelos.

### 3.2.4 *Modeling*

Esta es una etapa que trae muchos desafíos debido al gran volumen de datos que posee Mercadolibre, esto puede en algunos casos condicionar el uso de algunas técnicas de *machine learning* por el excesivo tiempo que llevaría el entrenamiento.

Los nuevos modelos generalmente se hacen utilizando Fury Data App (FDA) o AutoML (actualmente provisto por Google Cloud Platform). FDA es una herramienta interna que facilita la creación de modelos de *machine learning* permitiendo al científico de datos jugar con los hiperparámetros del modelo seleccionado. Si bien FDA da mucha mayor libertad al científico de datos para optimizar el modelo para el problema dado, tiene la contra de que la optimización de hiperparametros es un proceso inmaduro aún porque es manual, para automatizarlo en parte hay que usar scripts y *jupiter notebooks*. Otra diferencia entre ambos productos es el límite en cuanto a la cantidad de variables que puede tener un *Dataset*, en AutoML se permite hasta 1000 y en FDA no hay límites. Para algunos equipos esto es un problema pero para otros se puede trabajar sin inconvenientes con 1000 o menos variables. Dependiendo del problema y del criterio de cada equipo es que se usa mayoritariamente uno u otro o la comparación de ambos. En base al tiempo disponible es que se puede experimentar más buscando el mejor modelo posible o solo el que cumple con las métricas deseadas. En el caso de un reentrenamiento debe ser al menos mejor que el modelo actual.

Otro desafío es lo desbalanceado del problema, es decir, muy pocos casos de fraude vs la mayoría de transacciones buenas. Para resolver esta problemática se suele poner más peso a determinadas poblaciones del *dataset*, por ejemplo usuarios nuevos. Otra forma de tratar esto es haciendo *undersampling*, esto es cambiar la proporción de fraudes en el *dataset* para que tenga un mayor porcentaje de transacciones con fraude y así aprender mejor estos patrones.

Hay scripts automáticos que corren para comparar métricas estándar sobre el *machine learning success criteria*. Estas son *Accuracy, Recall, Confusión Matrix, AUC y F1-*

*Score*, gráficos de distribución de riego para fraudes confirmados y para transacciones buenas.

Otra métrica que se utiliza para elegir el mejor modelo es el % de bloqueo de transacciones necesario para detectar un determinado % de fraude, así mismo para un determinado % de bloqueo que % de aprobación final queda. Un ejemplo de ésto sería tener un 10% de bloqueo para detectar un 80% del fraude confirmado y esto deja una aprobación final del 70%. La aprobación final también depende de la aprobación bancaria. En algunas ocasiones si el modelo no funciona como se esperaba se analiza el motivo y eso puede que lleve a encontrar errores en etapas previas, tener que corregir y generar un nuevo *dataset* y modelo. Con estas métricas se decide cuál de los modelos pasa a la etapa de evaluación en donde se mira el *business success criteria.*

El tiempo requerido para esta etapa puede variar desde 3 días para algo muy simple hasta pudiendo llegar a un mes en caso de dedicar un buen tiempo a la experimentación y con *datasets* grandes.

El principal problema de esta etapa es que dado un *dataset* no se puede generar otro con las variantes necesarias sin hacer una nueva extracción. En caso de necesitar modificarlo hay que volver a generar el *dataset* con lo que eso implica.

### 3.2.5 *Evaluation*

En la etapa de evaluación se busca verificar que el modelo cumpla con el *business success criteria* que se definió inicialmente. Para esto algunos equipos hacen documentación que se completa con cada modelo, la más usada es un RFC (*Request For Change*) en donde se especifica en un lenguaje de negocio el motivo del nuevo modelo, los *features* que incorpora y las métricas de modelo en las diferentes etapas pre-productivas. Este RFC tiene un aprobador de negocio y uno de IT (uno diferente al que hizo el modelo) para validar que las métricas cumplen con el *business success criteria* y así subir a producción el modelo.

Una de las mayores dificultades de esta etapa es vincular el *business success criteria* con el *machine learning success criteria,* es decir, dadas las métricas de *machine learning* cuales serán los valores productivos de aprobación y fraude. Para obtener las métricas del RFC se corren otros scripts donde se le especifica cuánto se quiere frenar en fraude y eso optimiza los cortes de modelo por rango de monto. En estos scripts se especifica la aprobación mínima deseada, el mínimo de fraude que se quiere frenar y los rangos de montos de las transacciones en los cuales encontrar el punto de corte óptimo para las métricas deseadas. La salida del script especificará para cada rango de monto el umbral de riesgo a utilizar para bloquear una transacción si es superado. Un ejemplo, en el rango de 10 a 25 USD se debería bloquear las transacciones con un riesgo mayor a 0,9 y para el rango de 25 a 50 USD si el riesgo es mayor a 0,8. Con estos cortes escalonados se construyen las métricas del RFC con las cuales se hablará con negocio antes de salir a producción. Estos scripts se corren inicialmente con el *dataset* de validación y testeo pero también con *dataset* de transacciones recientes para evitar diferencias entre métricas de laboratorio y las productivas. El RFC además de incluir los cortes escalonados, especifica el % de bloqueo de transacciones, el % de fraude frenado, el % de aprobación final esperado, comportamiento del modelo en segmentos de interés (ej usuarios nuevos o electrónica) y una comparación contra el mejor modelo productivo.

En caso de no aprobarse el RFC el modelo tendrá que volver a etapas anteriores en busca de los motivos por los cuales no se cumplen las métricas esperadas. Esto puede llevar a incorporar nuevas características, corregir errores en el *dataset* o en el entrenamiento del modelo.

Otra documentación que se realiza en algunas ocasiones que requieran de mucha experimentación es un MDR (*Model Decision Request*) en donde se reflejan las diferentes decisiones técnicas que se fueron tomando en la experimentación del modelado y los resultados obtenidos.

En general la duración de esta etapa es de 1 día para obtener las métricas y documentación necesarios y hacer una evaluación por parte de los revisores del RFC

### 3.2.6 *Deployment*

El plan de *deployment* tiene un esquema similar en todos los equipos pero con algunas pequeñas variaciones dependiendo del contexto.

Inicialmente se pone el modelo en el ambiente productivo pero solo guardando las predicciones y sin accionar en base a ellas, llevando entre 1 y 2 días. Esto se hace como una última verificación de que todo está funcionando acorde a las métricas calculadas en etapas anteriores.

Posteriormente se activa el modelo en producción con una participación del 20% o 30%, otros equipos con un 5/10 % (100% si fuera el primer modelo de un producto o segmento). Se verifica en los días siguientes algunos casos que el modelo nuevo aprueba y los anteriores rechazan para detectar tempranamente falencias del modelo.

Entre 2 y 4 semanas (dependiendo de la prioridad del flujo, volumen de casos, etc) después se verifican las métricas reales del modelo antes de continuar aumentando su participación hasta llegar a la deseada. Es raro que un solo modelo tenga más del 70% de tráfico ya que la multiplicidad de modelos potencia la prevención de fraude.

Si en algunas de estas partes del plan de *deploy* no se observan las métricas esperadas se le quitará participación productiva al modelo hasta entender la causa raíz. Esto podría llevar nuevamente el modelo a etapas más tempranas.

En cuanto a monitoreo existen diferentes *dashboards* y alarmas para detectar incrementos de fraude en algún segmento, bajas en aprobación y otros indicadores que pueden anticipar posibles ataques. Los próximos modelos a hacer o reentrenar se determinarán en base a necesidades de negocio y las métricas que tengan esos modelos. Es normal que en flujos más prioritarios los entrenamientos ocurran más seguidos aunque las métricas sean saludables.

Una Oportunidad es el proceso de *deploy*, este tiene muchos pasos y algunos son difíciles de entender para un DS teniendo que agregar varias configuraciones en diferentes lugares y todo tiene que estar bien interconectado para que funcione correctamente.

Se han presentado en la presente sección como Mercadolibre implementa cada una de las 6 etapas de **CRISP-DM.** En base a las entrevistas con personas claves también

surgieron problemas o dolores en algunas de las etapas. En la siguiente sección se resumen cada uno de ellos.

## 3.4 Análisis de resultados de las entrevistas con los expertos.

En las entrevistas a expertos se les presentó el proceso que siguen los diferentes equipos. En el capítulo 2.2 se describió el proceso en sus seis etapas (*business understanding, data understanding, data preparation, modeling, evaluation y deployment*) y en el capítulo 3.2 la implementación de dicho proceso por parte de Mercadolibre en prevención de fraude. En dicha implementación se encontraron falencias o desafíos que son las mismas que ven los expertos, las cuales se detallan a continuación.

- ***Data Understanding***
  - Cuando se quiere incorporar una nueva variable o *feature*, si esta no se encuentra dentro de las fuentes de datos de fraude es probable que el DS o DE (*data engineer*) tenga que hacer algún programa ad-hoc en python o java para extraer esos datos y ver su poder predictivo. Esto es totalmente manual y propenso a errores. Las múltiples fuentes de datos y sus diversos formatos complejizan los análisis.
  - Falta de documentación sobre qué información tiene cada dato, su calidad y disponibilidad. Esto tanto para fuentes de datos internas como externas.
- ***Data Preparation***
  - Múltiples tecnologías para creación de *features*. Esto complejiza la creación y mantenimiento de estos.
  - Los *dataset* no permiten modificaciones ante un error que se deba corregir y eso lleva a que la extracción tenga que efectuarse nuevamente con lo que implica en tiempo y costos.
  - En caso de necesitar recrear un *dataset* puede haber demoras de varias horas. Un caso donde se hace más notorio el problema es cuando se tiene que entrenar un modelo con urgencia usando un *dataset* pequeño y se producen demoras debido a otras extracciones de gran volumen.
- ***Modeling***

- Algunos equipos usan solamente FDA para la creación de sus modelos y comparaciones entre diferentes modelos creados en FDA para encontrar el óptimo. Otros equipos hacen generalmente sus modelos con *automl* y comparan contra otros modelos hechos en *automl* o FDA. En general los equipos que suelen usar *automl* usan menor cantidad de *features* en comparación con los que solo usan FDA.

- ***Evaluation***
  - Es un desafío estimar con precisión el *business success criteria* desde el *machine learning success criteria*.
- ***Deployment***
  - La puesta en producción de *features* y asignación de tráfico a un modelo requiere que un DS o DE tenga que tocar varias configuraciones en diversos lugares, pudiendo provocar errores.

De acuerdo a los dolores o desafíos presentados los expertos expresaron una serie de recomendaciones para mitigarlos o eliminarlos.

1. ***Data Understanding***

   1.1. Ambos expertos coincidieron en la importancia de que todos los equipos de los diferentes productos de Mercadolibre puedan volcar al *data lake* la mayoría de su información. Esto ocurre en la mayoría de los casos pero no en todos. La recomendación para el corto plazo es proveer herramientas que permitan garantizar la calidad de los datos en las diferentes fases de *data understanding* para no tener que llegar a etapas más avanzadas y darse cuenta que algo está mal. Estas herramientas deberían facilitar la conexión y extracción de cualquier fuente de datos de la compañía como así también hacer verificaciones tempranas de la calidad del dato y no solamente tardías como ocurre cuando el *dataset* ya esta armado por medio de FraudTK.

   Uno de los expertos sugirió que a largo plazo sería bueno implementar a nivel compañía un *Data Mesh* que soluciona muchos de los problemas de datos que se detectaron. Uno de los 4 principios de este enfoque

propuesto es que los datos son tratados como producto brindando accesibilidad y calidad a los mismos.

1.2. Una adecuada documentación sobre la semántica de cada dato, sus tipos, los equipos que los mantienen, etc es parte fundamental de la mirada de datos como producto, como se mencionó esto es unos de los principios de *Data Mesh.*

Una solución más simple, sin necesidad de implementar un *Data Mesh*, es tener un buen catálogo de datos donde el consumidor pueda obtener la información necesaria sobre el dato a utilizar.

2. ***Data Preparation***

2.1. Uno de los expertos comentó que el verdadero problema no es la variedad de tecnologías para construir los *features* sino la falta de una interfaz unificada para todos ellos. Algunos van a requerir de diferentes tecnologías porque apuntan a problemas distintos, con tiempos de respuesta requeridos diferentes, lo que sí es importante es la interfaz unificada de cara al DS.

La recomendación de los expertos fue utilizar un *feature store* para la creación y gestión de los *features* utilizados por los modelos. Esto implica ir deprecando algunas soluciones e ir migrandolas al *feature store*. Actualmente el equipo trabaja con un *feature store* pero no todos los *features* están en el mismo. Los expertos recomendaron acelerar la migración para hacer un uso full de *feature store.* Esto simplificará notablemente la creación y mantenimiento de *features*.

2.2. Esta restricción de FraudTK se da para que el *dataset* se corresponda con la extracción inicial y así evitar que el vector de *features* con el que se entrenó sea diferente al que se termina usando en producción.

Uno de los expertos recomienda agregar la funcionalidad a FraudTK para permitir la modificación de *datasets*, sin tener que hacer una nueva extracción, pero llevando un registro de todos los cambios en el *dataset* para que a la hora de subir a producción se verifique que se cuenta con la misma definición del vector de *features* entre entrenamiento y producción.

Otro de los expertos sugirió que FraudTK debería hacer un *sampling* de los datos a extraer para mostrar al DS posibles errores o inconsistencias. El DS debería visualizar y confirmar la correctitud de los datos, posteriormente el proceso continúa con la extracción total.

2.3. Esto se produce debido a que cada solicitud de extracción es encolada para ser procesada asincrónicamente y al poseer solamente 2 *clusters,* pueden producirse retrasos cuando se arman grandes *datasets*. Los expertos coinciden en una solución que aumente la capacidad de procesamiento *on-demand*, pudiendo así procesar más en menor tiempo, pero asociando los costos al equipo que lo requiera. Esto implicaría que en determinados casos de urgencia se creen *clusters on-demand* para satisfacer la necesidad en ciertos casos de bajos tiempos de respuesta. De este modo se balancea rapidez y costos. Esto puede solucionar la necesidad de crear rápidamente *datasets* pequeños ante un ataque sin tener que esperar la culminación de otros.

3. ***Modeling***

3.1. Los expertos coincidieron en que no todos tienen que usar las mismas formas y herramientas para modelar porque los problemas pueden ser diferentes.

En la experiencia de los expertos los modelos con más *features* tienden a ser más estables ante fallas en datos o ataques y pueden aprender mayor cantidad de patrones con la contraparte de un entrenamiento menos simple.

*AutoML* tiene una gran potencia para optimizar los hiperparametros y encontrar una arquitectura de *machine learning* con muy buena performance pero con restricciones en cantidad de *features*. Esto último para muchos tipos de problemas no es un inconveniente. Ambos productos tienen fortalezas donde en determinadas situaciones puede ser mejor el uso de uno u otro.

Puede que para problemas donde ya se está buscando mejorar en modelos con muy buenos resultados, sea necesario modelos en FDA

donde el DS pueda aplicar su experiencia y sin restricciones de cantidad de *features* pueda lograr esos puntos extra de mejora. Aunque este último sea más artesanal puede valer la pena sobre todo en estos casos.

3.2. Uno de los expertos recomendó la profundización creación de modelos desatendidos para productos o poblaciones estables en cuanto a métricas. Un modelo desatendido es que el que entrena automáticamente con el mismo vector de *features* pero con nuevos datos para que aprenda nuevos patrones de comportamiento. Si las métricas de este nuevo modelo son las esperadas se procede a ponerlo en producción.

4. ***Evaluation***

4.1. Uno de los expertos recomendó que los objetivos de negocio se deben plasmar con mucha claridad y ser medibles ya que algunas veces no termina de estar claro cuál es la necesidad final, si es aprobación, reducción de fraude o trabajar en alguna población específica. En algunas ocasiones las métricas de *machine learning* se miden con una determinada temporalidad y las de negocio con otra, es importante que se midan con la misma frecuencia y con el mismo rango temporal. Otro de los expertos resaltó la importancia de entender el negocio para el que se va a modelar, entender bien cómo se comportan los usuarios buenos y los malos y como se refleja esto en los datos.

5. ***Deployment***

5.1. Ambos expertos coinciden en que las herramientas de asignación de tráfico a modelos y sus *features* en producción tienen que ir a un esquema de autoservicio evitando múltiples configuraciones en diferentes lugares. El *feature store* ayudará en los referente a productizar features y el desarrollo de herramientas orientadas al autoservicio facilitará la puesta en producción.

En la ilustración Nº 14 se presenta un cuadro que resume las oportunidades o dolores y sus respectivas recomendaciones.

*Ilustración 14: Resumen de oportunidades y recomendaciones.*

| **Etapa** | **Oportunidad** | **Recomendación** |
|---|---|---|
| Data Understanding | Programas *ad-hoc* para acceso a múltiples fuentes de datos | Corto plazo, herramientas que faciliten el acceso a diversas fuentes. En el largo plazo la implementación de *Data Mesh*. |
| Data Understanding | Falta de documentación de los datos | Un *Data Mesh* ya que este tiene los datos como producto y esto también implica documentación |
| Data Preparation | Múltiples tecnologías para la creación de *features* | Adopción full de un *feature store* |
| Data Preparation | *Datasets* inmutables | Agregar funcionalidad a FraudTK para permitir la modificación pero con el registros de las alteraciones para verificar el mismo vector de *features* en producción |
| Data Preparation | Generación de *datasets* con demoras | Creación de infra *on-demand* para casos de urgencia |
| Modeling | Diferentes herramientas para modelar entre equipos | En general cada equipo puede usar las herramientas que crea más adecuadas a su problema. Para algunos casos puede ser mejor una y en otro otra herramienta. |
| Evaluation | Estimar *business success criteria* desde el *machine learning success criteria*. | Claridad en la definición de poblaciones y sus métricas a alcanzar. Medición con misma temporalidad en ambos criterios. |
| Deployment | Asignación de tráfico productivo y configuración | Ir a un enfoque de autoservicio en la |

| Etapa | Oportunidad | Recomendación |
|---|---|---|
| | productiva de *features* | asignación de tráfico y adopción full de *feature store*. |

*Fuente: Elaboración propia*

# CONCLUSIONES

El objetivo de este trabajo ha sido describir el proceso de *machine learning* utilizado en la prevención de fraude en Mercadolibre y hacer aportes con sugerencias de mejoras en base a entrevistas con expertos.

De acuerdo a la investigación realizada en el marco teórico se pudo observar que Mercadolibre usa el proceso CRISP-DM descrito en el capítulo 2.2 para la creación de sus modelos de *machine learning*. Dicho proceso consta de 6 etapas, las cuales son:

1. *Business Understanding*
2. *Data Understanding*
3. *Data Preparation*
4. *Modeling*
5. *Evaluation*
6. *Deployment*

Tal como se presentó en el análisis de resultados se pudieron visualizar varios desafíos o falencias en algunas de las etapas del proceso de *machine learning.* Los más significativos son:

- ***Data Understanding***
  - Necesidad en algunas ocasiones de programas ad-hoc en python o java para extraer datos y ver su poder predictivo. Esto es totalmente manual y propenso a errores.
  - Múltiples fuentes de información y falta de documentación sobre los datos.
- ***Data Preparation***
  - Multiplicidad de tecnologías para la creación de *features* sin una interfaz común.
  - Demoras en la creación de algunos *datasets* críticos y la falta de herramientas que permitan modificar el mismo sin la necesidad de recrearlo.
- ***Deployment***
  - El proceso de asignación de tráfico productivo de un modelo requiere que un DS/DE tenga que tocar varias configuraciones en diversos lugares,

pudiendo provocar errores. Algo similar acontece para la configuración de *features* en producción.

Las recomendaciones para estas problemáticas o desafíos son:

- ***Data Understanding***
  - Para el corto plazo la recomendación es proveer herramientas que permitan conectarse de forma simple a diversas fuentes de datos permitiendo hacer validaciones de calidad de los datos

    Para el largo plazo un *Data Mesh* ayudaría en las diferentes problemáticas planteadas referentes a esta etapa.

- ***Data Preparation***
  - La necesidad de una interfaz común para la creación de *features* la industria la provee por medio de un *feature store*. La recomendación es mejorar el *feature store* y acelerar la migración al mismo. Esto mejorará problemas en esta etapa y en la de deployment.
  - La recomendación es una solución basada en costos fijos para el funcionamiento normal de la creación de *datasets* y otra con costos variables asignados a los equipos para casos de mayor urgencia.
- ***Deployment***
  - La rápida adopción total de un *feature store* mejorará la creación, mantenimiento y puesta en producción de los *features*. Asimismo es importante profundizar en un modelo de autoservicio simplificado para la asignación de tráfico productivo para los modelos.

# Bibliografía

Google & Atlas VPN. (23 de Marzo de 2020). Obtenido de https://atlasvpn.com/blog/google-registers-a-350-increase-in-phishing-websites-amid-quarantine/

Ceurvels, M. (14 de Diciembre de 2020). *Latin America will be the fastest-growing retail ecommerce market this year.* Obtenido de eMarketer: https://content-na2.emarketer.com/latin-america-will-fastest-growing-retail-ecommerce-market-this-year

Enrico, C. (23 de Abril de 2020). *El efecto de COVID-19 en el e-commerce.* Obtenido de Forbes Centroamerica: https://forbescentroamerica.com/2020/04/23/el-efecto-de-covid-19-en-el-ecommerce/

Radoini, A. (11 de Mayo de 2020). *Cyber-crime during the COVID-19 Pandemic.* Obtenido de UNICRI: http://www.unicri.us/news/article/covid19_cyber_crime

LexisNexis. (Enero de 2021). *The new cybercrime landscape.* Obtenido de LexisNexis: https://risk.lexisnexis.com/insights-resources/research/cybercrime-report

The Paypers. (Noviembre de 2020). *Fraud Prevention in ecommerce report 2020/2021.* Obtenido de The Paypers: https://www.europeanpaymentscouncil.eu/sites/default/files/inline-files/fraud-prevention-in-ecommerce-report-20202021.pdf

Columbus, L. (18 de Mayo de 2020). *How E-Commerce's Explosive Growth Is Attracting Fraud.* Obtenido de Forbes: https://www.forbes.com/sites/louiscolumbus/2020/05/18/how-e-commerces-explosive-growth-is-attracting-fraud/?sh=1f55ed1c6c4b

Cybersource. (2019). *2019 Global ecommerce fraud management report.* Obtenido de Cybersource: https://www.cybersource.com/content/dam/documents/en/global-fraud-report-2019.pdf

www.zdnet.com. (29 de Octubre de 2019). *Details for 1.3 million Indian payment cards put up for sale on Joker's Stash.* Obtenido de www.zdnet.com: https://www.zdnet.com/article/details-for-1-3-million-indian-payment-cards-put-up-for-sale-on-jokers-stash/

businesswire. (9 de Julio de 2020). *Kount Named #1 in Quadrant Evaluation of eCommerce Fraud Prevention Vendors.* Obtenido de Kount Named #1 in Quadrant Evaluation of eCommerce Fraud Prevention Vendors: https://www.businesswire.com/news/home/20200709005247/en/Kount-Named-1-in-Quadrant-Evaluation-of-eCommerce-Fraud-Prevention-Vendors

*NIST.* (1 de Agosto de 2007). Obtenido de https://www.nist.gov/: https://csrc.nist.gov/glossary/term/phishing

expertinsights. (25 de Marzo de 2021). *50 Phishing Stats You Should Know In 2021.* Obtenido de https://expertinsights.com: https://expertinsights.com/insights/50-phishing-stats-you-should-know/

Google. (3 de Mayo de 2021). *Machine learning: preguntas y respuestas.* Obtenido de Google: https://www.google.com/intl/es/about/main/machine-learning-qa/

Torralba, P. P. (13 de Abril de 2021). *¿Qué es el Machine Learning? Aprendizaje supervisado vs no supervisado.* Obtenido de iebschool: https://www.iebschool.com/blog/que-machine-learning-big-data/

Universidad de Alcalá. (1 de Enero de 2018). *¿EN QUÉ CONSISTE EL MACHINE LEARNING?* Obtenido de Universidad de Alcalá: https://www.master-data-scientist.com/machine-learning-data-science/

Gonzalez, J. L. (8 de Febrero de 2018). *Tipos de aprendizaje automático.* Obtenido de medium.com: https://medium.com/soldai/tipos-de-aprendizaje-autom%C3%A1tico-6413e3c615e2

APD. (4 de Marzo de 2019). *¿Qué es Machine Learning y cómo funciona?* Obtenido de apd.es: https://www.apd.es/que-es-machine-learning/

Instituto de Ingeniería del Conocimiento. (2 de Enero de 2021). *Aprendizaje profundo por refuerzo.* Obtenido de www.iic.uam.es: https://www.iic.uam.es/aprendizaje-profundo-por-refuerzo/

Martínez-Plumed, F., Contreras-Ochando, L., Ferri, C., Hernández Orallo, J., Kull, M., Lachiche, N., . . . Flach, P. (27 de Diciembre de 2019). CRISP-DM Twenty Years Later: From Data Mining Processes to Data Science Trajectories. *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, 1*(1), 1. Obtenido de https://ieeexplore.ieee.org/abstract/document/8943998/

Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., & Wirth, R. (2000). CRISP-DM 1.0 Step-by-step data mining guide. Obtenido de https://www.kde.cs.uni-kassel.de/wp-content/uploads/lehre/ws2012-13/kdd/files/CRISPWP-0800.pdf

Jupyter. (1 de Enero de 2021). *Jupyter.* Obtenido de Jupyter: https://jupyter.org/

Mishra, A. (24 de Febrero de 2018). *Metrics to Evaluate your Machine Learning Algorithm.* Obtenido de https://towardsdatascience.com/: https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234

Infobae. (15 de June de 2021). *Estafa por WhatsApp: el falso mensaje viral que ofrece productos gratis por el aniversario de Mercado Libre.* Obtenido de Infobae.com: https://www.infobae.com/economia/2021/06/15/estafa-por-whastapp-el-falso-mensaje-viral-que-ofrece-productos-gratis-por-el-aniversario-de-mercado-libre/

Arteaga, S. (27 de February de 2017). Ataque phishing te roba las claves de Netflix y los datos bancarios. *ComputerHoy.com*. Obtenido de https://computerhoy.com/noticias/software/ataque-phishing-te-roba-claves-netflix-datos-bancarios-58982

Narkhede, S. (26 de June de 2018). *Understanding AUC - ROC Curve | by Sarang Narkhede.* Recuperado el 3 de January de 2022, de Towards Data Science: https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5

IBM. (s.f.). *IBM Cloud Docs.* Recuperado el 3 de January de 2022, de IBM Cloud Docs: https://cloud.ibm.com/docs/knowledge-studio?topic=knowledge-studio-evaluate-ml

Saurav Singla. (31 de October de 2020). *What is AutoML and Why is it important?* Recuperado el 3 de March de 2022, de Towards Data Science: https://towardsdatascience.com/what-is-automl-6ddf27040f27

Dehghani, Z. (20 de May de 2019). *How to Move Beyond a Monolithic Data Lake to a Distributed Data Mesh.* Obtenido de Martin Fowler: https://martinfowler.com/articles/data-monolith-to-mesh.html

Tiagi, A., Ananthakrishnan, H., Carrero, I. P., & Lakshminarayan, K. (26 de July de 2021). *Data Movement in Netflix Studio via Data Mesh | by Netflix Technology*

*Blog.* Obtenido de Netflix TechBlog: https://netflixtechblog.com/data-movement-in-netflix-studio-via-data-mesh-3fddcceb1059

FeatureStore.org. (s.f.). *Feature Store for ML - What is a Feature Store?* Obtenido de Feature Store: https://www.featurestore.org/what-is-a-feature-store

Cimpanu, C. (13 de July de 2020). *A hacker is selling details of 142 million MGM hotel guests on the dark web.* Recuperado el 9 de May de 2022, de ZDNet: https://www.zdnet.com/article/a-hacker-is-selling-details-of-142-million-mgm-hotel-guests-on-the-dark-web/

Merchants Savvy. (Octubre de 2020). Obtenido de Merchants Savvy: https://www.merchantsavvy.co.uk/payment-fraud-statistics/

Holds, D. (Junio de 2020). *Databricks.com*. Obtenido de Databricks.com: https://databricks.com/session_na20/building-a-real-time-feature-store-at-ifood

Tecton. (23 de Febrero de 2021). *Tecton.ai*. Obtenido de Tecton.ai: https://www.tecton.ai/blog/atlassian-accelerates-deployment-of-ml-models-from-months-to-days-with-tecton/

Dataops.live. (s.f.). *Dataops.live*. Obtenido de Dataops.live: https://www.dataops.live/case-studies-roche?hsLang=en

Mercadolibre.com. (2022). *Mercadolibre INC - Form 10k - 2021.*

# ANEXO 1: PREGUNTAS PARA LAS ENTREVISTAS PERSONAS CLAVES

Las preguntas realizadas a personas claves de los equipos de prevención de fraude de Mercadolibre fueron:

- ¿ El proceso de machine learning que siguen tiene correspondencia con CRISP-DM (Martínez-Plumed et al., 2019, 2) ?
- Descripción de cada una de las etapas del proceso
- Especificar los mayores dolores o trabas que experimentan en cada etapa al momento de hacer machine learning.

# ANEXO 2: PREGUNTAS PARA LAS ENTREVISTAS CON LOS EXPERTOS

Las entrevistas con los expertos fueron guiadas por las siguientes preguntas:

- Explicar el proceso de machine learning que siguen los equipos y sus dolores.
- En base a lo antes expuesto ¿ Que recomendaciones les darías para eliminar o mitigar esos dolores ?
- ¿ Otra recomendación que no esté relacionada con sus problemas pero que les permita mejorar el proceso a fin de producir modelos con mayor velocidad y precisión ?