https://doi.org/10.69639/arandu.v12i3.1542

Modelo de aprendizaje automático para evaluar vulnerabilidad de ingeniería social en usuarios de internet del Tecnológico de Estudios Superiores Chalco

A Machine Learning Model for Assessing Social Engineering Vulnerability among Internet Users at the Technological Institute of Higher Studies of Chalco

Raúl Romero Castro
raul_rc@tesch.edu.mx
https://orcid.org/0000-0002-7450-9278
Tecnologico de Estudios Superiores Chalco
México

José Daniel Ruperto Villalpando
jose_rv1@tesch.edu.mx
https://orcid.org/0009-0002-7691-8435
Tecnologico de Estudios Superiores Chalco
México

Dulce Arisbeth Córdoba Beltrán
dulce_cb1@tesch.edu.mx
https://orcid.org/0009-0000-7274-4875
Tecnologico de Estudios Superiores Chalco
México

Fabián Soberanes Martín
fabian_sm@tesch.edu.mx
https://orcid.org/0000-0002-5876-3161
Tecnologico de Estudios Superiores Chalco
México

Guadalupe Nayeli Villanueva Valdivia
guadalupe_vv@tesch.edu.mx
https://orcid.org/0000-0002-9773-3563
Tecnologico de Estudios Superiores Chalco
México

Artículo recibido: 18 agosto 2025 - Aceptado para publicación: 28 septiembre 2025
Conflictos de intereses: Ninguno que declarar.

RESUMEN

El desarrollo de un modelo auxiliar enfocado en el análisis de la ciberdelincuencia en el Tecnológico de Estudios Superiores de Chalco, busca delimitar los niveles de vulnerabilidad presentes, proponiendo un tema de relevancia para la población, particularmente en el análisis de riesgos y vulnerabilidades relacionados con técnicas de ingeniería social, las cuales exponen a los usuarios a situaciones desfavorables en la protección de sus datos personales, especialmente la Información Personal Identificable (PII). Una vez que las personas interactúan con el modelo, comienzan a generar conciencia acerca del nivel de riesgo y de la necesidad de fortalecer sus

medidas de seguridad digital. El modelo diseñado para procesar esta información es impulsado por técnicas de Machine Learning y para su desarrollo se seleccionan características esenciales, como el aprendizaje supervisado, empleando algoritmos de regresión logística múltiple que permiten la clasificación del conjunto de datos suministrado, garantizando un análisis estructurado y confiable. El objetivo principal consiste en examinar un dataset específico para generar predicciones precisas y resultados interpretables. La metodología contempla la recolección, limpieza, tratamiento y preparación de datos, seguida de la implementación del algoritmo de regresión logística múltiple, lo que permite modelar relaciones complejas entre múltiples variables independientes y una variable dependiente categórica.

Palabras clave: aprendizaje automático, ingeniería social, vulnerabilidad, algoritmo, regresión lineal

ABSTRACT

The development of an auxiliary model focused on the analysis of cybercrime at the Tecnologico de Estudios Superiores de Chalco seeks to delimit existing levels of vulnerability, addressing a topic of significant relevance for the population, particularly in the assessment of risks and vulnerabilities associated with social engineering techniques, which expose users to unfavorable situations regarding the protection of their personal data, especially Personally Identifiable Information (PII). Once individuals interact with the model, they begin to develop awareness of their risk level and the need to strengthen digital security measures. The model designed to process this information is driven by Machine Learning techniques, and for its development, essential characteristics are selected, such as supervised learning, employing multiple logistic regression algorithms that enable the classification of the supplied dataset, ensuring a structured and reliable analysis. The main objective is to examine a specific dataset in order to generate accurate predictions and provide interpretable results. The methodology involves the collection, cleaning, processing, and preparation of data, followed by the implementation of the multiple logistic regression algorithm, which allows for modeling complex relationships between multiple independent variables and a single categorical dependent variable.

Keywords: machine learning, social engineering, vulnerability, algorithm, linear regression

INTRODUCCIÓN

Los ataques digitales afectan a los usuarios y organizaciones cada día y continúan creciendo de manera exponencial. Dentro de este tipo de acciones, existen métodos como la Ingeniería Social y el Phishing, los cuales se aprovechan de errores humanos para robar información por medio de técnicas de manipulación, engaño, o suplantación de identidad. El impacto de la ingeniería social y sus técnicas, como método de robo de datos a través de medios digitales después de la COVID-19 así como los riesgos asociados a este crecimiento, incluyendo la posibilidad de una pérdida de datos masiva, han puesto en alerta al público, instituciones cibernéticas y grupos de investi- gación, para que estos estén constantemente difundiendo a su alrededor consejos, prácticas útiles a realizar, encuestas y demás actividades, tratando de reducir el núme- ro de incidentes que suceden día con día.

Las recientes investigaciones de organizaciones de ciberseguridad, han estado combatiendo estás prácticas negativas para mitigar los riesgos, usando protocolos de seguridad que ayudan a prevenir que intrusos y ataques a los sistemas puedan afectar redes, aplicaciones y computadoras para que estos estén mejor protegido. En el año 2019, cuando el CORONAVIRUS, aislaba al mundo del contacto físico, los hackers y practicantes de ingeniería social, continuaron adaptándose a la situación, llevando sus procesos delictivos y adaptándolos al entorno digital, llevando el robo de datos al siguiente nivel. En México, los ciberataques se centraron en el ámbito banca- rio, como es el caso de Vishing, similar al Phishing, a diferencia de que este es reali- zado a través de llamadas telefónicas, donde los atacantes se hacen pasar por una empresa legítima para poder robar información confidencial. Los ataques de Malware y Ransomware también son frecuentes en México. Estos programas maliciosos pue- den infectar computadoras y dispositivos móviles, cifrar archivos y exigir un rescate para su liberación.

En este año los casos de ciberataques en México, tras la pandemia de COVID- 19, hubo un crecimiento en delitos cibernéticos que pasaron de apenas 300.3 millones en el 2019 a 120 mil millones de intentos en el 2021, un crecimiento de casi 400 veces, lo que convirtió al país en el más atacado en América Latina. Según los resultados presentados en el sitio web Pulso Capital (Pulso Capital, 2023).

METODOLOGÍA

Aprendizaje supervisado

Es cuando entrenamos un algoritmo de Machine Learning dándole las preguntas (características) y las respuestas (etiquetas). Así en un futuro el algoritmo pueda hacer una predicción conociendo las características. En este tipo de aprendizaje hay dos algoritmos (entrenamientos): el de clasificación y el de regresión.

Algoritmo de clasificación: esperamos que el algoritmo nos diga a qué grupo pertenece el elemento en estudio. El algoritmo encuentra patrones en los datos que le damos y los clasifica

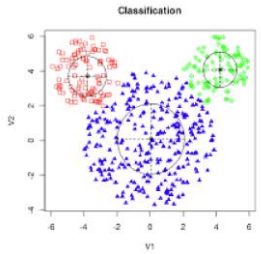
en grupos. Luego compara los nuevos datos y los ubica en uno de los grupos y es así como puede predecir de que se trata.

La variable por predecir es un conjunto de estados discretos o categóricos. Pueden ser:

- *Binaria: {Sí, No}, {Azul, Rojo}, {Fuga, No Fuga}, etc.*
- *Múltiple: Comprará {Producto1, Producto 2…}, etc.*
- *Ordenada: Riesgo {Bajo, Medio, Alto}, etc.*

Imagen 1

Predictores

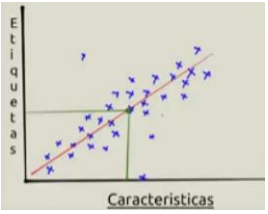


Fuente: http://redicces.org.sv/jspui/bitstream/10972/3626/1/Art6_RT2018.pdf

Algoritmo de regresión: en este método lo que se espera es un número. No lo ubica en un grupo, sino que devuelve un valor específico

Imagen 2

Regresión



Fuente: http://redicces.org.sv/jspui/bitstream/10972/3626/1/Art6_RT2018.pdf

Modelos de Machine Learning

Los algoritmos de Machine Learning, se pueden agrupar en tres modelos:

1. **Modelos lineales** Estos tratan de encontrar una línea que se "ajuste" bien a la nube de puntos que se disponen. Aquí destacan desde modelos muy conocidos y usados como la regresión lineal (también conocida como la regresión de mínimos cuadrados), la logística (adaptación de la lineal a problemas de clasificación -cuando son variables discretas o categóricas-). Estos dos modelos tienen el problema del "overfit", esto significa que se ajustan "demasiado" a los datos disponibles, con el riesgo que esto tiene para nuevos datos que pudieran llegar. Al ser modelos relativamente simples, no ofrecen resultados muy buenos para comportamientos más complicados
2. **Modelos de árbol** Son modelos precisos, estables y más sencillos de interpretar básicamente porque construyen unas reglas de decisión que se pueden representar como un árbol. A diferencia de los modelos lineales, pueden representar relaciones no lineales

para resolver problemas. En estos modelos, destacan los árboles de decisión y los random forest (una media de árboles de decisión). Al ser más precisos y elaborados, obviamente ganamos en capacidad predictiva, pero perdemos en rendimiento

3. **Redes neuronales** Las redes artificiales de neuronas tratan, en cierto modo, de replicar el comportamiento del cerebro, donde tenemos millones de neuronas que se interconectan en red para enviarse mensajes unas a otras. Esta réplica del funcionamiento del cerebro humano es uno de los "modelos de moda" por las habilidades cognitivas de razonamiento que adquieren.

Aprendizaje no supervisado

Aquí solo le damos las características al algoritmo, nunca las etiquetas. Queremos que agrupe los datos que le dimos según sus características. El algoritmo solo sabe que como los datos comparten ciertas características, de esa forma asume que pueda que pertenezcan al mismo grupo.

El desarrollo del algoritmo destinado a nuestro sistema de aprendizaje automático, estará basado en la regresión logística múltiple, la cual Rodrigo, J. A. en su website Cienciadedatos.net, establece lo siguiente: La regresión logística múltiple es una extensión de la regresión logística simple. Se basa en los mismos principios que la regresión logística simple (explicados anteriormente) pero ampliando el número de predictores. Los predictores pueden ser tanto continuos como categóricos.

Imagen 3

Regresión logística

$$\ln\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_i x_i$$

$$logit(Y) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_i x_i$$

Fuente: https://cienciadedatos.net/documentos/27_regresion_logistica_simple_y_multiple

El valor de la probabilidad de *Y* se puede obtener con la inversa del logaritmo natural:

Imagen 4

Probabilidad de Y

$$p(Y) = \frac{e^{\beta_0+\beta_1 x_1+\beta_2 x_2+\ldots+\beta_i x_i}}{1 + e^{\beta_0+\beta_1 x_1+\beta_2 x_2+\ldots+\beta_i x_i}}$$

Fuente: https://cienciadedatos.net/documentos/27_regresion_logistica_simple_y_multiple

A la hora de evaluar la validez y calidad de un modelo de regresión logística múltiple, se analiza tanto el modelo en su conjunto como los predictores que lo forman. Se considera que el modelo es útil si es capaz de mostrar una mejora respecto al modelo nulo, el modelo sin predictores.

Una vez definido el algoritmo a utilizar, es indispensable preparar los datos con los cuales el modelo será alimentado, por tanto el contar con una buena estructura de limpieza de datos es

esencial, como lo menciona Brownlee, L. En su libro *Data preparation for machine learning: Data cleaning, feature selection, and data transforms in Python,* "La limpieza de datos es un paso de vital importancia en cualquier proyecto de aprendizaje automático. En los datos tabulares, existen muchos análisis estadísticos diferentes y técnicas de visualización de datos que puede utilizar para explorar sus datos a fin de identificar las operaciones de limpieza de datos que desee realizar". (Brownlee, 2020).

Método

El desarrollo del modelo está basado en la metodología CRISP-DM, como menciona Schröer, C. En la obra: A systematic literature review on applying CRISP-DM, Las recomendaciones de la guía del usuario de CRISP-DM se han utilizado principalmente en las fases que van desde la comprensión del negocio hasta la evaluación. Existen diferencias en la estructura y en la forma en que se describen las tareas específicas." (Schröer, C, 2021).

Esta metodología es usada en proyectos de ciencia de datos, que según Udacity. En la obra: CRISP-DM explained: a proven data mining methodology. Menciona que "Es un marco ampliamente adoptado que describe los pasos involucrados en un proyecto de análisis de datos o ciencia de datos. Su objetivo principal es proporcionar un enfoque sistemático, garantizando que los proyectos estén bien definidos, gestionados y produzcan resultados valiosos". (Udacity, 2025) La metodología cuenta con seis fases de desarrollo:

Entendimiento del negocio

- **Entendimiento de datos**
- **Preparación de datos**
- **Construcción del modelo**
- **Evaluación del modelo**
- **Despliegue del modelo**
- **Entendimiento del negocio**

Primer paso de la metodología, donde se establece el diseño del modelo, así como los alcances de este. En el modelo, se establecen objetivos como son el de conocer los niveles de vulnerabilidad de la población estudiantil del Tecnológico, entre otros.

Entendimiento de los datos

Diseño del formulario que servirá de base para la creación del dataset que será utilizado en múltiples tareas dentro del desarrollo del modelo.

Preparación de los datos

Como menciona Oracle en la obra, Process overview: machine learning for SQL use cases (CRISP-DM methodology), menciona lo siguiente: "La fase de preparación implica finalizar los datos y cubre todas las tareas involucradas en la elaboración de los datos en un formato que pueda utilizar para construir el modelo.

Es probable que las tareas de preparación de datos se realicen varias veces, de forma iterativa y sin ningún orden prescrito. Las tareas pueden incluir la selección de columnas (atributos), así como la selección de filas en una tabla". (Oracle, 2021).

Diseño del Dataset

Para el desarrollo del dataset que servirá para alimentar y a su vez entrenar al modelo, se optó por crearlo a partir de una encuesta (Google Form) diseñada con el fin de obtener información acerca de las técnicas de ingeniería social a las cuales la población ha sido expuesta. El uso de la herramienta Google Form ayudó a que la difusión sea más práctica para la población a analizar debido a que es una herramienta de fácil uso y conocida por muchos.

Una vez difundido el formulario, los datos son tratados y separados de la población total a sólo 113 individuos. Los cuales se usarán para delimitar los 3 posibles perfiles de resultado para el modelo, los perfiles se mencionan posteriormente.

Tratamiento de los datos

Se refiere a identificar y corregir errores en los datasets que pueden impactar de forma negativa a un modelo predictivo o de clasificación.

Como señalan Lima, C. A. F., Luz, B. M., Takemoto, S. T., Barisson, P. Jr., Tezzin, R. A. T., Peres, L. E. A., dos Santos, F. N. G. M., Anarelli, T. N., & da Silva, A. F. (2015). En la obra titulada: strategic modeling for the characterization of the conditions that allow the anticipation of the consumer's requests. "La preparación de los datos y la organización del modelado requirieron de toda la estructura de caracterización del big data y de las reglas de gobernanza. Esto se logró mediante la creación de un conjunto de índices que permitieran el acceso y la categorización de los datos originales (sin procesar o primarios), así como la consolidación previa a la compilación realizada". (Lima et al., 2015).

Al igual que como menciona Wirth, R., & Hipp, J. En la obra: CRISP-DM: Towards a standard process model for data mining, menciona que "La fase de preparación de datos abarca todas las actividades para construir el conjunto de datos final (datos que se incorporarán a las herramientas de modelado) a partir de los datos brutos iniciales. Es probable que las tareas de preparación de datos se realicen varias veces, sin seguir un orden preestablecido. Estas tareas incluyen la selección de tablas, registros y atributos, la limpieza de datos, la creación de nuevos atributos y la transformación de datos para las herramientas de modelado." (Wirth, R., & Hipp, J. 2008)

Existen muchos tipos de errores que podrían existir en un dataset, aunque algunos de los errores más comunes incluyen errores con columnas que no contienen información suficiente o donde existen filas duplicadas. Para el dataset, estos no representan problema alguno, debido a que los datos fueron normalizados al momento de su recolección.

Construcción del modelo

Según el autor MyEducator, en la obra The Data Mining Process – CRISP-DM steps, menciona que los modelos son la aplicación de algoritmos para buscar, identificar y mostrar cualquier patrón o mensaje en sus datos. Hay dos tipos básicos de modelos en minería de datos: aquellos que clasifican y aquellos que predecir. (MyEducator, s.f.)

Herramientas de construcción del modelo

El modelo de aprendizaje automático destinada para el modelo, fue creada en la herramienta de Google Colab donde se crea la libreta de Júpiter notebook, en el lenguaje de programación Python.

Diseño del modelo de aprendizaje automático

El desarrollo del modelo es a través del lenguaje de programación Python, pues este ofrece librerías especializadas para el desarrollo de modelos de aprendizaje automático. Para el desarrollo, se optó por Numpy, Pandas y Skelearn, debido a que son las librerías que más se ajustaban al desarrollo y ofrecían funciones y métodos a utilizar.

Selección de la población

La delimitación de la población-muestra, es establecida a 113 respuestas, las cuales fueron analizadas y normalizadas para su correcto uso y con el fin de tener un dataset estructurado para que el modelo pueda ser alimentado y que esté a su vez no genere errores debido a un sesgo.

Aplicación de terciles

Posterior a la separación de la muestra, esta es a su vez separada en terciles, los cuales ayudarán a la delimitación de los posibles perfiles que el modelo compare para poder mostrar una respuesta.

Balanceo de datos

El balanceo de datos en modelos de aprendizaje automático hace referencia a el estado donde una cantidad de muestras es exactamente la misma, es decir, que la distribución de las clases de los conjuntos de datos del dataset este equilibrada, esto con el fin de que el modelo, no esté orientado o se incline a dar un resultado teniendo otras variables.

Como menciona el autor Miranda, J. V. En la obra: Cómo lidiar con el desbalanceo de datos, menciona que: Al entrenar un modelo de clasificación con la variable no balanceada, encontraremos algunos problemas. Esto sucede porque el patrón de datos de la clase dominante superará a los de la clase con menos frecuencia. Generalmente, en bases de datos que tienen una variable objetivo-desbalanceada, la clase con la frecuencia más baja es precisamente la que nos interesa predecir, lo que hace que los problemas sean aún mayores. Miranda, J. V. (2023).

Etiquetas del nivel de vulnerabilidad

Las etiquetas de nivel de vulnerabilidad, son establecidas a partir de tres perfiles hipotéticos acerca de las posibles respuestas. A partir de las respuestas en el formulario, se establecen los niveles de vulnerabilidad.

Dentro de los niveles establecidos y de qué forma fueron clasificados se encuentran:

Niveles de vulnerabilidad

- Baja, con un rango de 0 a 10 puntos en el formulario.
- Moderada, con un rango de 11 a 18 puntos en el formulario.
- Alta, con un rango de 19 a 31 puntos en el formulario.

Evaluación del modelo

Según el autor Data Science PM. En la obra CRISP-DM for Data Science, menciona que: "la fase de evaluación del modelado se centra en la evaluación técnica del modelo, la fase de evaluación analiza de forma más amplia qué modelo se adapta mejor al negocio y qué pasos seguir." (Data Science PM, 2024).

Al finalizar el desarrollo del modelo, el siguiente paso es testearlo en busca de errores, los principales errores son debido a que, al momento de ser ejecutado el modelo, indique errores de programación. Otro error común, es que el modelo desarrollado comience a dar resultados fuera de los parámetros de entrenamiento.

Despliegue del modelo

Como menciona el autor Shearer, C. En la obra, el modelo CRISP-DM: el nuevo plan para la minería de datos. Journal of Data Warehousing, menciona que: Dependiendo de los requisitos, la fase de implementación puede ser tan simple como generar un informe o tan compleja como implementar un proceso de minería de datos repetible en toda la empresa. Un modelo no es particularmente útil a menos que el cliente pueda acceder a sus resultados. (Shearer, C. 2000).

Una vez finalizado el testeo y determinado su correcto funcionamiento, el modelo aún no será publicado o difundido para su uso, puesto que se considera que este aún se encuentra en una fase experimental, donde aún existen características a pulir.

RESULTADOS Y DISCUSIÓN

El término ingeniería social, acuñado por primera vez en la informática fue por Kevin Mitnick, quien sostiene que la Ingeniería Social se refiere a la aplicación de técnicas, que los hackers utilizan para engañar a un usuario autorizado de sistemas informáticos de una compañía para que revele información sensitiva, o para lograr que de forma insospechada realice acciones que creen un hueco de seguridad que pueda ser explotado.

Una vez que el término tomó relevancia entre los especialistas y la población, el crecimiento de está y el desarrollo de nuevas técnicas de la misma índole, hicieron que la seguridad informática juegue un papel importante con el objetivo de asegurar que los datos almacenados en nuestros ordenadores se mantengan libre de cualquier peligro.

Según el autor Cristian Borghello menciona que el "arte de engañar" puede ser utilizado por cualquiera, desde un vendedor que se interesa en averiguar las necesidades de sus compradores para ofrecerles un servicio, hasta creadores de malware y atacantes que buscan que

un usuario revele su contraseña de acceso a un determinado sistema. Más allá de las coincidencias, o no, en el límite de lo éticamente correcto, todo intento de obtener información confidencial para un uso inapropiado, resulta una actividad altamente cuestionable.(Borghello C, 2009).

Fragmento del código del modelo de aprendizaje automático

```
# Configurar permisos de Drive en Colab
from google.colab import drive
drive.mount('/content/drive')
# Importar las librerias
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
# Cargar datos
data = pd.read_excel('/content/drive/MyDrive/Colab Notebooks/Proyecto_Ingenieria_Social/encuesta_2024_para_poblacion_113.xlsx', skiprows = [1])
data
# Seleccionar variables independientes (X) y dependiente (y)
# Independientes
X = data[['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z', 'aa', 'ab']]
# Dependiente y variable objetivo
y = data['nivel']
# Dividir los datos en conjuntos de entrenamiento y prueba
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
# Entrenar el modelo de regresión logística
model = LogisticRegression()
model.fit(X_train, y_train)
# Realizar predicciones
y_pred = model.predict(X_test)
# Calcula la precisión
accuracy = accuracy_score(y_test, y_pred)
# Calcula la matriz de confusión
conf_matrix = confusion_matrix(y_test, y_pred)
# Reporte de clasificación
```

```
class_report = classification_report(y_test, y_pred)
print(f"Accuracy: {accuracy}")
print("Confusion Matrix:")
print(conf_matrix)
print("Classification Report:")
print(class_report)
# Función que solicita nuevos valores de características al usuario
def solicitar_nuevas_caracteristicas():
# Predecir el nuevo target usando las características proporcionadas por el usuario
nuevas_caracteristicas = solicitar_nuevas_caracteristicas()
nueva_prediccion = model.predict(nuevas_caracteristicas)
#Despues del despliegue de las preguntas, el modelo determina el nivel de vulnerabilidad y arroja finalmente el resultado
print(f"Tu nivel de vulnerabilidad ante técnicas de Ingeniería Social es:  {nueva_prediccion[0]}")
```

Métrica de resultados

Los resultados obtenidos del modelo se dividen en tres métricas principales **accuracy, matriz de confusión** y **classification report**, que incluye la precisión (**precision**), la capacidad de detección (**recall**) y la puntuación F1 (**f1-score**).

ACCURACY (PRECISIÓN)

Este valor indica qué tan bien ha clasificado el modelo los ejemplos de prueba correctamente. En tu caso:

Accuracy = 0.826 o **82.6%**, lo que significa que el modelo ha clasificado correctamente aproximadamente el 82.6% de los ejemplos de prueba.

Imagen 5

Precisión

```
[25] # Calcula la precisión
     accuracy = accuracy_score(y_test, y_pred)
     print(f"Accuracy: {accuracy}")

     Accuracy: 0.8260869565217391
```

Fuente: Elaboración propia.

Matriz de confusión

La matriz de confusión permite evaluar cuántas predicciones del modelo se han realizado de manera correcta y cuántas de forma errónea

Imagen 6

Matriz de confusión

```
[26] # Calcula la matriz de confusión
     conf_matrix = confusion_matrix(y_test, y_pred)
     print("Confusion Matrix:")
     print(conf_matrix)

     Confusion Matrix:
     [[3 0 1]
      [0 8 1]
      [0 2 8]]
```

Fuente: Elaboración propia.

Donde cada fila corresponde a las etiquetas verdaderas (reales), y cada columna corresponde a las predicciones del modelo, para este caso el modelo cuenta con tres clases de vulnerabilidades: ALTA; MODERADA y BAJA.

Primera fila (ALTA):

- El modelo predijo correctamente 3 casos como **ALTA**.
- No cometió errores prediciendo como **BAJA**.
- Cometió 1 error prediciendo como **MODERADA** cuando en realidad eran **ALTA**.

Segunda fila (BAJA):

- El modelo predijo correctamente 8 casos como **BAJA**.
- Cometió 1 error prediciendo como **MODERADA** en lugar de **BAJA**.

Tercera fila (MODERADA):

- El modelo predijo correctamente 8 casos como **MODERADA**.
- Cometió 2 errores prediciendo como **BAJA** cuando en realidad eran **MODERADA**.

Reporte de clasificación

En este reporte se incluyen las metricas de **precision** (precisión), **recall** (sensibilidad) y **f1-score** para cada clase.

Imagen 7

Reporte de clasificación

```
# Reporte de clasificación
class_report = classification_report(y_test, y_pred)
print("Classification Report:")
print(class_report)

Classification Report:
              precision    recall  f1-score   support

        ALTA       1.00      0.75      0.86         4
        BAJA       0.80      0.89      0.84         9
    MODERADA       0.80      0.80      0.80        10

    accuracy                           0.83        23
   macro avg       0.87      0.81      0.83        23
weighted avg       0.83      0.83      0.83        23
```

Fuente: Elaboración propia

a. Precision (Precisión)

La precisión indica cuántas de las instancias predichas como una clase particular son realmente de esa clase.

- **ALTA**: 1.00 (100% de las instancias predichas como ALTA realmente son ALTA).
- **BAJA**: 0.80 (El 80% de las instancias predichas como BAJA realmente son BAJA).
- **MODERADA**: 0.80 (El 80% de las instancias predichas como MODERADA realmente

son MODERADA).

b. Recall (Sensibilidad o Exhaustividad)

El **recall** mide la capacidad del modelo para encontrar todas las instancias reales de una clase particular. Es decir, cuántas veces predijo correctamente cuando realmente pertenecía a esa clase.

- **ALTA**: 0.75 (El modelo identificó correctamente el 75% de las instancias que realmente eran ALTA).
- **BAJA**: 0.89 (El modelo identificó correctamente el 89% de las instancias que realmente eran BAJA).
- **MODERADA**: 0.80 (El modelo identificó correctamente el 80% de las instancias que realmente eran MODERADA).

c. F1-score

El **F1-score** es la media armónica entre la precisión y el recall. Proporciona una medida equilibrada de precisión y recall, especialmente cuando las clases están desbalanceadas.

- **ALTA**: 0.86.
- **BAJA**: 0.84.
- **MODERADA**: 0.80.

d. Soporte (support)

El soporte es el número de instancias reales de cada clase en los datos de prueba:

- **ALTA**: 4 instancias.
- **BAJA**: 9 instancias.
- **MODERADA**: 10 instancias.

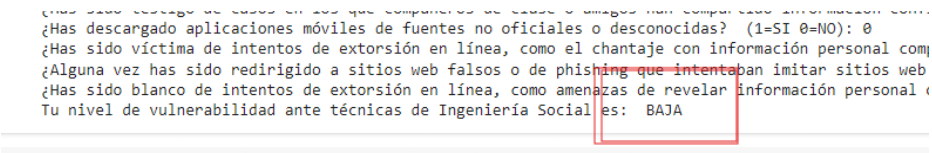
Interpretación de Resultados

En el modelo se implementó el algoritmo de clasificación, regresión logística múltiple, para mejorar el desempeño en cuanto a precisión se refiere sobre el aprendizaje automático. Así como los posibles niveles de vulnerabilidad los cuales fueron explicados anteriormente.

Las siguientes interpretaciones son sobre la ejecución del modelo en el cuaderno de jupyter notebook, en la herramienta de Google Colab. Resultados, del modelo, donde a partir de las respuestas generadas, se determina el nivel de vulnerabilidad. Caso de nivel de vulnerabilidad “BAJA”.

Imagen 5

Caso de vulnerabilidad BAJA

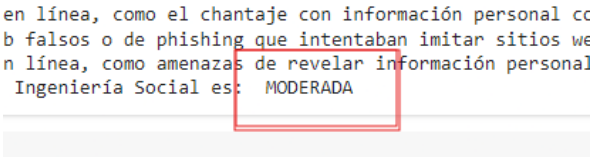


Fuente: Elaboración propia

Caso de nivel de vulnerabilidad “MODERADA”

Imagen 6

Caso de vulnerabilidad MODERADA

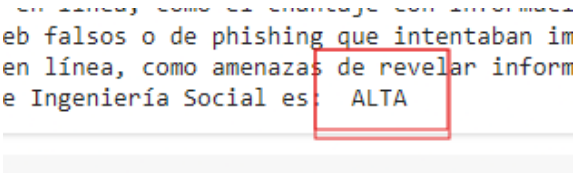


Fuente: Elaboración propia

Caso de nivel de vulnerabilidad “ALTA”

Imagen 7

Caso de vulnerabilidad ALTA



Fuente: Elaboración propia

CONCLUSIONES

El desarrollo de modelos de aprendizaje automático tienen un amplio espectro de aplicaciones en distintos ámbitos, en el caso del modelo desarrollado, contribuyen en gran manera a proteger datos y sistemas contra diversas amenazas, como es en la detección de intrusiones, filtrado de Spam y Phishing, siendo este donde más se relaciona el modelo, así como también a la autentificación y control de acceso.

El desarrollo del modelo de aprendizaje automático, destinado a conocer el nivel de vulnerabilidad ante ingeniería social de los individuos de la zona oriente del Estado de México, crea conciencia y marca una pauta a orientar la ciberseguridad hacia las zonas de población no céntricas y en casos rurales, para poder salvaguardar datos importantes de la ciudadanía.

Así como también, a la implementación de nuevas medidas de seguridad, por parte de la población. Finalmente, el desarrollo del modelo motiva a que otros desarrolladores comiencen a interesarse en esta área y en estas zonas de población.

Respecto a la precisión del modelo presentado tiene un rendimiento aceptable, con una accuracy del 82.6% y métricas bastante equilibradas.

REFERENCIAS

Borghello, A. C., & De abril del, E. M. de E. P. L. F. L. 13. (s.f.). El arma infalible: la Ingeniería Social. Eset-la.com. Recuperado el 25 de abril de 2024, de https://www.eset-la.com/pdf/prensa/informe/arma_infalible_ingenieria_social.pdf

Brownlee, J. (2020). Data preparation for machine learning: Data cleaning, feature selection, and data transforms in Python. [s.l.]: Machine Learning Mastery.

Data Science PM. (2025). CRISP-DM for Data Science. Recuperado el 31 de agosto de 2025, de https://www.datascience-pm.com/wp-content/uploads/2024/12/CRISP-DM-for-Data-Science-2025.pdf

Grande, C. E. L. (s.f.). Ingeniería Social: El Ataque Silencioso. Org.sv. Recuperado el 25 de abril de 2024, de http://www.redicces.org.sv/jspui/bitstream/10972/2910/1/Articulo6.pdf

Lima, C. A. F., Luz, B. M., Takemoto, S. T., Barisson, P. Jr., Tezzin, R. A. T., Peres, L. E. A., dos Santos, F. N. G. M., Anarelli, T. N., & da Silva, A. F. (2015). Strategic modeling for the characterization of the conditions that allow the anticipation of the consumer's requests. Open Journal of Social Sciences, 3(10), 36–47. https://doi.org/10.4236/jss.2015.310005

Miranda, J. V. (2023). Cómo lidiar con el desbalanceo de datos. Alura Cursos. https://www.aluracursos.com/blog/como-lidiar-con-el-desbalanceo-de-datos

MyEducator. (s.f.). The Data Mining Process – CRISP-DM steps. Recuperado el 31 de agosto de 2025, de https://app.myeducator.com/reader/web/1539k/blah1/j83yi/

Oracle. (2021). Process overview: Machine learning for SQL use cases (CRISP-DM methodology). Recuperado el 12 de agosto de 2025, de https://docs.oracle.com/en/database/oracle/machine-learning/oml4sql/21/mlsql/process-overview.html

Pulso, C. (2023). México entre los 10 países con más ciberataques. Recuperado el 19 de abril de 2024, de https://pulsocapital.com/mexico-entre-los-10-paises-con-mas-ciberataques/

Rodrigo, J. A. (2016, agosto). Regresión logística simple y múltiple. Cienciadedatos.net. Recuperado el 30 de mayo de 2024, de https://cienciadedatos.net/documentos/27_regresion_logistica_simple_y_multiple

Sandoval, L. J. (2018, diciembre). Algoritmos de aprendizaje automático para análisis y predicción de datos. Repositorio Digital de Ciencia y Cultura de El Salvador (REDICCES). Recuperado el 30 de mayo de 2024, de http://redicces.org.sv/jspui/bitstream/10972/3626/1/Art6_RT2018.pdf

Schröer, C. (2021). A systematic literature review on applying CRISP-DM. Procedia Computer Science. Recuperado el 12 de junio de 2025 de ScienceDirect.

Shearer, C. (2000). El modelo CRISP-DM: el nuevo plan para la minería de datos. Journal of

Data Warehousing, 5(5), 13–22. Recuperado de https://www.datascience-pm.com/crisp-dm-2/

Udacity. (2025, marzo). CRISP-DM explained: a proven data mining methodology. Recuperado el 28 de agosto de 2025 de UDacity Blog.

Wirth, R., & Hipp, J. (2000). CRISP-DM: Towards a standard process model for data mining. DaimlerChrysler AG. Recuperado de https://cs.unibo.it/~danilo.montesi/CBD/Beatriz/10.1.1.198.5133.pdf