Revista Innovación y Software
Vol. 3, No. 1, Mes Marzo-Agosto, 2022
ISSN: 2708-0935
Pág. 6-16
https://revistas.ulasalle.edu.pe/innosoft

Tipo de artículo: Artículos originales
Temática: Inteligencia artificial
Recibido: 05/09/2021 | Aceptado: 08/11/2021 | Publicado: 30/03/2022

Identificadores persistentes:
ARK: ark:/42411/s8/a49
PURL: 42411/s8/a49

# Aplicación de los árboles de decisión en la identificación de sitios web fraudulentos

# *Application of decision trees in the identification of fraudulent websites*

**Christian Layme Fernández [1], José Manuel Suri Canaza [2], David Jose Peña Ugarte [3], Jhon Yoset Luna Quispe [4]**

[1] Universidad Nacional de San Agustín. Arequipa, Perú. claymef@unsa.edu.pe
[2] Universidad Nacional de San Agustín. Arequipa, Perú. jsurica@unsa.edu.pe
[3] Universidad Nacional de San Agustín. Arequipa, Perú. dpenau@unsa.edu.pe
[4] Universidad Nacional de San Agustín. Arequipa, Perú. jlunaq@unsa.edu.pe

[*] Autor para correspondencia: claymef@unsa.edu.pe

**Resumen**

La seguridad informática, es un área muy importante en cualquier sistema que tenga conexión a internet, debido a que existen sitios Web fraudulentos que pueden realizar acciones delictivas hacia una persona, organización u otra entidad. Por lo cual es necesario poder detectar qué sitios web son fraudulentos antes de poder ingresar a ella, para ello se desarrolló una implementación mediante Árboles de Decisión con el lenguaje de Python para poder detectar y clasificarlos en Legítimos, Sospechosos y Fraudulentos por medio de 1353 casos que clasifican a los sitios webs.

**Palabras clave:** Árbol de Decisión, Python, Seguridad Informática, Web Sites.

***Abstract***

*Computer security is a very important area in any system that has an internet connection, because there are fraudulent websites that can carry out criminal actions towards a person, organization or other entity. Therefore, it is necessary to be able to detect which websites are fraudulent before being able to enter it, for this an implementation was developed through Decision Trees with the Python language to detect and classify them as Legitimate, Suspicious and Fraudulent through 1353 cases that they rank websites.*

***Keywords:*** *Decision Tree, Python, Computer Security, Web Sites.*

## Introducción

Uno de los cambios más notorios actualmente en el mundo, aún más por el tiempo de cuarentena, es el uso de Internet y de servicios Web, como son el comercio electrónico, el cual aumentó, por las circunstancias que atraviesa el mundo, Internet se convirtió no solo en un gran apoyo para la sociedad, que utiliza el *E-commerce* para poderse suministrar las necesidades que tenga, Internet también está siendo utilizado por delincuentes que roban la información de muchas personas y cometiendo actos fraudulentos, el cual genera pérdidas de miles de millones cada año, estos sitios web suelen presentarse como sitios web amigables y de fuentes legítimas de información, productos y servicios en línea.

Los sitios web fraudulentos tienen características, como son: ***SFH***, Si el sitio web tiene definido la propiedad action en sus formularios de forma correcta dirigiendo los datos de los formularios a direcciones del mismo sitio web. ***PopUp Windows***, Si el sitio tiene ventanas emergentes (*PopUp*). ***SSL Final State***, Si el sitio web utiliza conexión *SSL*, que la misma presente un estado final válido. ***Request URL***, cuando son objetos (imágenes, scripts, hojas de estilos) que son cargadas de otra *URL* distinta a la del sitio. ***URL of Anchor***, cuando los objetos de una página son cargados desde el mismo sitio o desde un subdominio del mismo. ***Web Traffic*** Si tiene configurado un analizador de tráfico web como el *Google Analytics*. ***URL Length*** Cantidad de caracteres de la dirección URL. ***Age of Domain***, Cantidad de años que lleva activo el dominio de la URL. ***Have IP***, Si tiene o no dirección IP. Las medidas existentes, que disponen los buscadores, han mejorado pero no son suficientes para detectar sitios web fraudulentos, aún más por el esfuerzo de los que generan este tipo de sitios web para poder eludir estas medidas.

El uso de Árboles de Decisión resulta ser una buena opción para poder detectar los sitios web fraudulentos. Un árbol de decisión es una forma gráfica y analítica de representar todos los eventos (sucesos) que pueden surgir a partir de una decisión asumida en cierto momento. Nos ayudan a tomar la decisión “más acertada”, desde un punto de vista probabilístico, ante un abanico de posibles decisiones. Permite desplegar visualmente un problema y organizar el trabajo de cálculos que deben realizarse.

El propósito de este artículo, es el desarrollo un detector de sitios web fraudulentos, a través de inteligencia artificial usando árboles de decisión, con uso librerías para desarrollar el aprendizaje en *Python*, teniendo 1353 casos en los que se clasifican los sitios web, que serán evaluados y empleando la herramienta *Graphviz* para visualizar el Árbol decisión resultante.

## Herramientas

### Dataset

Es un conjunto de datos conocido también por el anglicismo *dataset*, es una colección de datos habitualmente tabulada. En general, y en su versión más simple, un conjunto de datos corresponde a los contenidos de una única tabla de base de datos o una única matriz de datos estadística, donde cada columna de la tabla representa una variable en particular, y cada fila representa a un miembro determinado del conjunto de datos en cuestión [1].

La publicación de los conjuntos de datos usados en un experimento son clave para su reproducibilidad, y cada vez son más las leyes públicas y normas de revistas científicas que obligan a hacerlos públicos, para evitar sesgos. Existen muchas fuentes de *datasets* como *kaggle*, el mismo *google* tiene una plataforma para buscar *dataset* "*Dataset Search*" [2].

### Colab

*Colaboratory* es un proyecto de investigación de Google creado para ayudar a difundir la educación y la investigación sobre *Machine Learning*. Es un entorno de *Jupyter* que no requiere configuración para su uso y se ejecuta completamente en la nube. [3]

Funcionamiento

*Colaboratory* es gratuita y forma parte de la suite de aplicaciones de Google en la nube. Por ello, para utilizarlo basta con acceder a nuestra cuenta de Google y, o bien entrar directamente al enlace de *Google Colab* o ir a nuestro *Google Drive*, pulsar el botón de «Nuevo» y desplegar el menú de «Más» para seleccionar «*Colaboratory*», lo que creará un nuevo cuaderno (notebook) [4].

Revista Innovación y Software
Vol. 3, No. 1, Mes Marzo-Agosto, 2022
ISSN: 2708-0935
Pág. 6-16
https://revistas.ulasalle.edu.pe/innosoft

Entorno de Ejecución

Cada celda es independiente, pero todas las celdas en un cuaderno utilizan el mismo *kernel*. El *kernel* es el motor de computación que está por debajo y que se encarga de ejecutar nuestro código y devolver el resultado para mostrarlo en la celda. El estado del *kernel* persiste durante el tiempo, con lo que, aunque cada celda sea independiente, las variables declaradas en una celda se pueden utilizar en las demás celdas [4].

**Librerías**

```
from pandas import Series, DataFrame
import pandas as pd
```

Figura 1. Librería *pandas*

Pandas es una herramienta de manipulación de datos de alto nivel desarrollada por Wes McKinney. Es construido con el paquete *Numpy* y su estructura de datos clave es llamada el *DataFrame*. El *DataFrame* te permite almacenar y manipular datos tabulados en filas de observaciones y columnas de variables [5].

```
import numpy as np
```

Figura 2. Librería *numpy*

*Numpy* es la biblioteca central para la computación científica en Python. Proporciona un objeto de matriz multidimensional de alto rendimiento y herramientas para trabajar con estas matrices.[6].

```
import os
import matplotlib.pylab as plt
```

Figura 3. Librería *matplotlib*

*Matplotlib* es una biblioteca de trazado 2D de Python que produce figuras de calidad de publicación en una variedad de formatos impresos y entornos interactivos en todas las plataformas. *matplotlib* se puede usar en scripts de Python, el *shell* de *python* e *ipython*, servidores de aplicaciones web y seis juegos de herramientas de interfaz gráfica de usuario.

*matplotlib* intenta hacer las cosas fáciles y las difíciles posibles. Puede generar gráficos, histogramas, espectros de potencia, gráficos de barras, gráficos de error, gráficos de dispersión, etc., con solo unas pocas líneas de código [7].

```
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import classification_report
import sklearn.metrics
```

Figura 4. Librería *matplotlib*

*Scikit-learn* anteriormente *scikits.learn* y también conocido como *sklearn* es una biblioteca de *machine learning* de software libre para el lenguaje de programación Python. [9] Cuenta con varios algoritmos de clasificación, regresión y agrupamiento que incluyen *support-vector machines*, *random forests*, *gradient boosting*, *k-means* y *DBSCAN*, y está diseñado para interoperar con las bibliotecas numéricas y científicas de *Python NumPy* y *SciPy*, sus principales algoritmos realizan [8]:

- *Classification*: Identifica a qué categoría pertenece un objeto
- *Regression*: Predice un atributo de valor continuo asociado con un objeto.
- *Clustering*: Agrupación automática de objetos similares en conjuntos.
- *Dimensionality reduction*: Reduce el número de variables aleatorias a considerar.
- *Model selection*: Comparara, válida y elige parámetros y modelos.
- *Preprocessing*: Extrae y normaliza las características.

## Métodos y Metodología computacional

Se usó para el desarrollo de este detector de sitios web fraudulentos usamos el modelo CRISP-DM, la cual comprende de seis fases: análisis del problema, análisis de Datos, preparación de los Datos, modelado, evaluación y explotación [10].

- La fuente de información fue un Dataset.csv, esto fue sacado de internet con datos reales, para poder construir un árbol de decisión utilizando las librerías *sklearn*, y se hizo uso de las funciones que nos brinda.

- Análisis del problema: Identificamos qué tipos de clasificación tendrán las páginas que se van a probar, viendo que pueden ser tres, y también identificamos la información necesaria para poder hacer esta clasificación.
- Análisis de Datos: con el *dataset* que obtuvimos, vimos que cantidad de información tiene, y las variables lingüísticas las clasificamos en números para su uso correcto con la librería que se trabajaría.
- Preparación de Datos: Se hizo una limpieza del *dataset*, con las funciones que nos brinda la libre, como eliminación de campos vacíos, datos irrelevantes.
- Modelado: Seleccionamos la técnica adecuada para poder hacer la clasificación.
- Evaluación: Tuvimos que corroborar efectivamente que el modelo escogido se ajuste a lo que estamos buscando, en este caso poder clasificar los diferentes sitios web.

## Resultados y discusión

Teniendo en cuenta las fases de la metodología *CRISP-DM* los siguientes son los resultados de cada una de las etapas:

### Análisis del problema

La Dirección de Seguridad Informática de una universidad pretende desarrollar un sistema informático capaz de detectar si un sitio web es fraudulento.

### Análisis de los datos

Del *dataset* descargado de internet, después de consultar el número de casos de las tablas de hechos para los nueve eventos en cuestión como se muestra en la Figura 5, se cuenta con un número de 1353 datos por cada categoría. Sin embargo, no todos los eventos comparten las mismas características. Una primera modificación de atributos se realizó para estandarizar aquellas variables lingüísticas dentro del *dataset*. El número de eventos modificado es 7, y son los siguientes: *SFH, popUpWindows, SSLfinalState, Request URL, URL of Anchor,web Traffic,Have IP* y Clasificación del sitio web. Con el fin de obtener los atributos estandarizados para el estudio, se realizó una búsqueda y análisis inicial de errores de los atributos, cambiando la codificación utilizando el rango establecido entre -1, 0 y 1. Basado en la información recepcionada se selecciona *Clasificación del sitio web* como variable objetivo. Como resultado se obtuvo el repositorio inicial con los 1353 datos, y sirvió de base para las siguientes etapas de este proceso.


Figura 5. Mapa de burbujas de los rasgos a evaluar

**Preparación de los datos**

Se realizó un análisis de la calidad de los datos del *dataset*, donde se identificó por cada atributo el número de valores distintos, el número de valores nulos, el valor máximo, valor mínimo, media y moda. Como resultado de este proceso, se seleccionaron los 9 eventos, sobre el cual se aplicaron las técnicas de modelado. Todos estos atributos son características propias del tema evaluado.

**Modelado**

Con el fin de encontrar o descubrir sitios web fraudulentos, se seleccionó el modelo de clasificación basado en árboles de decisión. La clasificación con árboles de decisión considera clases disjuntas, de forma que el árbol conducirá mediante sus hojas a la predicción diseñada. Después de construido el modelo servirá para determinar en nuevos casos, el tipo de sitio web visitado. Para esta tarea, se escogió como clase el atributo clasificación que determina si el sitio web es fraudulenta, sospechosa, o legítima. Para evaluar la calidad del modelo, se dividió el repositorio de datos en dos

conjuntos: entrenamiento y prueba. Se empleó el 75% de los datos para el entrenamiento del modelo, mientras que el 25% restante se utilizó para la prueba. Los grupos de datos antes explicados se toman de forma aleatoria en cada iteración. Se calculó el error de muestra parcial del modelo por cada iteración. Por último, se construyó el modelo con todos los datos. Para la lectura de los resultados de la evaluación se utiliza la matriz de confusión. Se visualiza una matriz de confusión tomada de un ejemplo aleatorio del proyecto. Además de la matriz, se visualiza la precisión del árbol, todo esto se muestra en la Figura 6.

| Predicted | -1 | 0 | 1 | All |
|---|---|---|---|---|
| True | | | | |
| -1 | 149 | 4 | 18 | 171 |
| 0 | 5 | 16 | 7 | 28 |
| 1 | 14 | 6 | 120 | 140 |
| All | 168 | 26 | 145 | 339 |

0.8407079646017699

Figura 6. Matriz de confusión y precisión del árbol

**Evaluación**

Analizando los resultados de las pruebas de clasificación con árboles de decisión realizadas con el conjunto de datos, en el cual se almacenan los datos de 1353 casos, se puede observar que el árbol de decisión construido con un factor de entrenamiento del 75%, esto quiere decir 1014 casos tomados aleatoriamente, y un número de casos de evaluación correspondiente al 25% que representa a los 339 casos restantes, con 285 instancias correctamente clasificadas, que corresponden a un porcentaje de precisión del 84,07% y 54 instancias incorrectamente clasificadas, correspondiente a un porcentaje de error del 15,93%. El árbol de decisión generado evalúa cada caso buscando que la entropía se iguale a cero. En algunas ramas, los resultados se visualizan en capas superiores, y de esta manera, mientras se aumenta la

profundidad del árbol se encuentra mayor cantidad de resultados. El árbol de decisión es de tipo binario, ya que solo muestra dos ramas hijas, algunas concluyendo en hojas de manera temprana. El algoritmo genera la repartición y categorización de los nodos hijos haciendo una evaluación ponderada, utilizando teoría de árboles de decisión, y brindando un número que sirve de particionador en nodo hijo izquierdo y nodo hijo derecho, así sucesivamente hasta llegar al final. Cada nodo hoja evalúa la partición de cada evento, además de su entropía, mostrando también el número de datos evaluados y mostrando cuantos pertenecen a cada sección de la clasificación (fraudulento=-1, sospechoso=0 y legítimo=1).

**Implementación**

Teniendo en cuenta los patrones descubiertos, es importante implementar planes de prevención que permitan disminuir los sitios web fraudulentos. Implementar programas de de ciber seguridad que permitan prevenir la infiltración delictiva. De igual manera, se deben redoblar esfuerzos en educación de seguridad cibernética y sanciones más enérgicas contra los infractores.

Este conocimiento descubierto se incorporará al existente y se integrará a los procesos de toma de decisiones de los analizadores de sitios web fraudulentos, proporcionando nuevos panoramas.

## Discusión

Según la matriz de confusión de la Figura 2, el modelo clasifica correctamente al 43,95% de casos fraudulentos, al 4,72% de casos sospechosos y al 35,40% de casos legítimos. Por otra parte, la precisión con la clase real de este modelo es de 0,84 que se considera aceptable (1,0 significa que ha habido precisión absoluta). Los porcentajes de instancias correctamente clasificados presentados en el árbol como en la matriz de confusión indican que el modelo tiene una precisión buena, por lo tanto, es confiable y eficiente, para clasificar nuevos casos, especialmente fraudulentos y legítimos. Este estudio da resultados aceptables que servirán de base al estudio futuro de reconocimiento de sitios web fraudulentos.

## Conclusiones

Los resultados obtenidos con el modelo de clasificación por árboles de decisión, indican que este es capaz de generar modelos consistentes con la realidad observada y el respaldo teórico, basándose únicamente en los datos que se

encuentran almacenados en el *dataset*. Los porcentajes de instancias correctamente clasificadas presentados en el árbol como en la matriz de confusión muestran que el modelo tiene una precisión buena y por consiguiente es confiable para clasificar nuevos casos, especialmente de sitios web fraudulentos y legítimos. El factor de confianza sirve de respaldo a los porcentajes antes expuestos. Como trabajos futuros se plantea utilizar otros clasificadores para comparar estos resultados y mejorar la precisión (por ejemplo, regresión logística), especialmente en los sitios web fraudulentos y aumentar el porcentaje de evaluación de sospechosos; aplicar tareas descriptivas de minería de datos como asociación y agrupación, con el fin de encontrar relaciones y similitudes, según el tipo de sitio web analizado. Dentro de esta investigación no hubo conflicto de intereses.

## Referencias

[1] "Find Open Datasets and Machine Learning Projects | Kaggle", Kaggle.com, 2020. [Online]. Available: https://www.kaggle.com/datasets. [Accessed: 12 Aug 2020].

[2]"Conjunto de datos", Es.wikipedia.org, 2020. [Online]. Available: https://es.wikipedia.org/wiki/Conjunto_de_datos#cite_ref-Editorial_1-0. [Accessed: 12- Aug- 2020].

[3] "Colab Notebooks", Magenta, 2020. [Online]. Available: https://magenta.tensorflow.org/demos/colab/. [Accessed: 12- Aug- 2020].

[4] "Google Colab: Python y Machine Learning en la nube - Adictos al trabajo", Adictos al trabajo, 2020. [Online]. Available: https://www.adictosaltrabajo.com/2019/06/04/google-colab-python-y-machine-learning-en-la-nube/. [Accessed: 12- Aug- 2020].

[5] "Pandas Basics - Learn Python - Free Interactive Python Tutorial", Learnpython.org, 2020. [Online]. Available: https://www.learnpython.org/es/Pandas%20Basics. [Accessed: 12- Aug- 2020].

[6] "Python Numpy Tutorial (with Jupyter and Colab)", Cs231n.github.io, 2020. [Online]. Available: https://cs231n.github.io/python-numpy-tutorial/#numpy. [Accessed: 12- Aug- 2020].

[7] S. Programacion en Castellano, "Introducción a la librería Matplotlib de Python", Programación en Castellano., 2020. [Online]. Available: https://programacion.net/articulo/introduccion_a_la_libreria_matplotlib_de_python_1599. [Accessed: 12- Aug- 2020]

[8] "scikit-learn: machine learning in Python — scikit-learn 0.23.2 documentation", Scikit-learn.org, 2020. [Online]. Available: https://scikit-learn.org/stable/. [Accessed: 12- Aug- 2020].

[9] Fabian Pedregosa; Gaël Varoquaux; Alexandre Gramfort; Vincent Michel; Bertrand Thirion; Olivier Grisel; Mathieu Blondel; Peter Prettenhofer; Ron Weiss; Vincent Dubourg; Jake Vanderplas; Alexandre Passos; David Cournapeau; Matthieu Perrot; Édouard Duchesnay (2011). "Scikit-learn: Machine Learning in Python". Journal of Machine Learning Research. 12: 2825–2830.

[10] J. Gallardo, “Metodología para el Desarrollo de Proyectos en Minería de Datos CRISP-DM” oldemarrodriguez.com, para. 2, Aug. 12, 2007. [Online]. Available: http://www.oldemarrodriguez.com/yahoo_site_admin/assets/docs/Documento_CRISP-DM.2385037. [Accessed Aug. 12, 2020].