Lab 12: Proyecto de Consultoría

Clustering

Luis R. Furlán

Octubre 2025

Python y Spark

A estas alturas están adquiriendo una reputación mundial por sus destrezas con ML.

Una “start-up” de tecnología en California requiere de su ayuda!

Python y Spark

Es hora de que vayan a San Francisco a ayudar a esta “start-up”.

Python y Spark

Esta “start’-up" ha sido recientemente “hackeada" y necesitan su ayuda para averiguar sobre los “hackeadores".

Python y Spark

Afortunadamente sus ingenieros forenses han logrado captar datos valiosos sobre los “hacks”, incluyendo información como la duración de la sesión, ubicaciones, velocidad de tecleado (wpm / palabras por minuto), etc.

Python y Spark

La ingeniero forense les informa lo que ha podido determinar hasta ahora. Ha capturado metadatos de cada sesión que los “hackeadores" utilizaron para conectarse a sus servidores.

Las siguientes son las características de los datos…

Python y Spark

- **'Session_Connection_Time'**: Cuánto duró la sesión en minutos
- **'Bytes Transferred'**: Cantidad de MB transferidos durante la sesión
- **'Kali_Trace_Used'**: Indica si el “hacker" estuvo usando Kali Linux
- **'Servers_Corrupted'**: Número de servidores corrompidos durante el ataque
- **'Pages_Corrupted'**: Número de páginas ilegalmente accedidas
- **'Location'**: Ubicación de donde provino el ataque (probablemente es info inútil ya que los “hackers" usaron VPNs)
- **'WPM_Typing_Speed'**: La velocidad estimada de tecleado basado en las bitácoras de las sesiones.

Python y Spark

La empresa tecnológica sospecha de 3 “hackers" potenciales que perpetraron el ataque.

Están seguros de los primeros dos pero no están seguros si el tercero estaba involucrado o no.

¡Solicitan su ayuda!

Python y Spark

¿Pueden ustedes ayudarles para determinar si el tercer sospechoso tuvo algo que ver con los ataques, o si fueron solamente dos?

Posiblemente no sea posible tener certeza completa, pero quizás lo que saben sobre Clustering les pueda ayudar.

Python y Spark

Una cosa más, la ingeniero forense tiene conocimiento de que los “hackers" se intercambian los ataques.

Esto quiere decir que cada uno de ellos debiera tener aproximadamente el mismo número de ataques.

Python y Spark

Por ejemplo, si hubieron un total de 100 ataques, entonces en una situación de 2 “hackers" cada uno debiera tener cerca de 50 ataques. En una situación de 3 “hackers" cada uno debiera tener unos 33 ataques.

Python y Spark

La Ingeniero piensa que éste es un elemento crucial en resolver esto, pero no sabe cómo distinguir/separar estos datos no-etiquetados en grupos de “hackers”.

Python y Spark

Suerte con este proyecto, ¡será divertido!

¡Disfrútenlo!