Trabajo Práctico 2 - Machine Learning

Cátedra Argerich

Trabajo Práctico 2 - Machine Learning

Competencia: *Binary Classification of VPN Proxy IP Address*.

(Tanto la participacion en la competencia como la resolucion del trabajo seran individuales).

Objetivo del trabajo

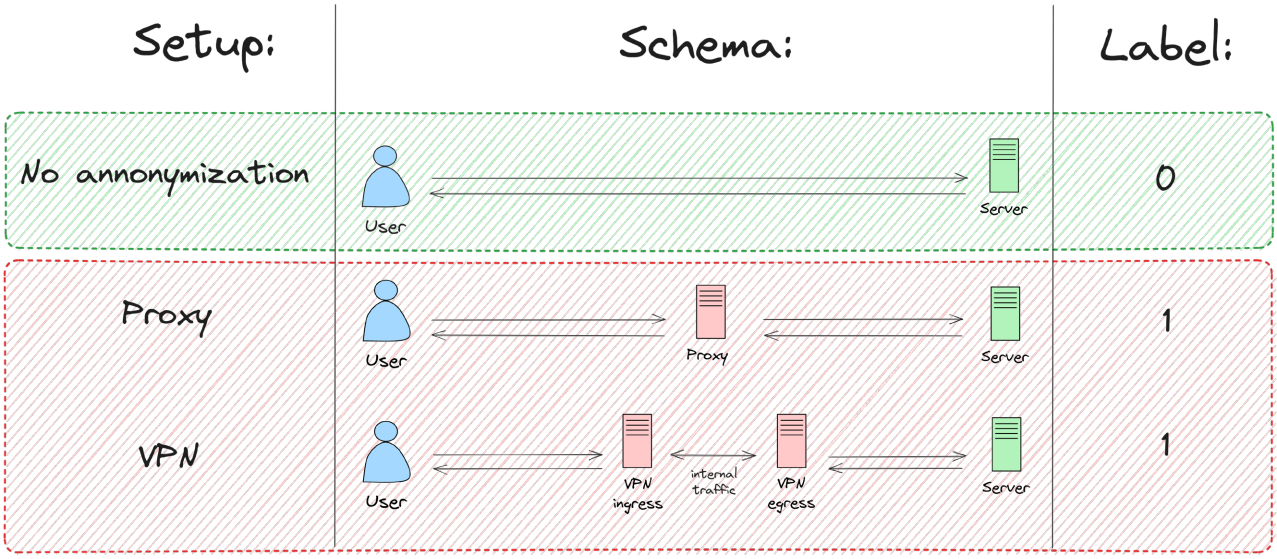
Su tarea es crear un modelo que pueda predecir si una dirección IP está asociada con un servicio de VPN o Proxy en el marco de una competencia de Kaggle. Los alumnos deberán participar en la competencia de Kaggle: *Binary Classification of VPN Proxy IP Address*.

Los modelos serán evaluados en función de su desempeño en el conjunto de pruebas privadas proporcionado por la plataforma. Si bien la participacion en la competencia es central, el resultado de ella no es determinante en la nota. Se debe prestar atención al criterio de corrección y a las consignas de esta pagina ya que son las que usaremos. En particular, el score que vamos a utilizar para validar nosotros es el mismo que el que usa la competencia: **F1-Score**.

Dataset a usar

El conjunto de datos proporcionado consiste en una muestra aleatoria de ataques reportados por el software de CrowdSec. Ambos sets (el de training y el de test), estan disponibles en la pagina de la competencia.

Cada entrada representa un ataque, y consta de algunas características del mismo (hora del ataque, tipo, etc) segun lo inferido por el sistemas de detección de amenazas. Debido a la naturaleza confidencial de los datos, las direcciones IP se encuentran anonimizadas en un ID compartido entre todos los archivos. Tengan en cuenta que el conjunto de datos está bastante desequilibrado, ya que solo alrededor del 5% de las entradas se identifican como procedentes de VPN o servidores proxy.



Parte I - Análisis exploratorio (6 Puntos)

Se deben realizar al menos 6 visualizaciones interesantes que **ayuden a explicar el target**. Deben incluirse al menos **un plot de cada tipo**:

- Bar plot
- Heatmap
- Plot 2D con distribucion marginal

Parte II - Baseline (4 Puntos)

Antes de realizar los dos modelos que tienen que entregar, deben realizar un tercer modelo muy sencillo que utilizar como baseline y tener de referencia. En general esta es una tarea muy importante que queremos que repitan en sus proyectos de machine learning. ¿Por qué?

- Navaja de Ockam: “Cuando se ofrecen dos o más explicaciones de un fenómeno, es preferible la explicación completa más simple; es decir, no deben multiplicarse las entidades sin necesidad.” ¿Para qué desarrollar un modelo super complejo si capaz es peor o casi igual que uno muy sencillo?

- Nos sirve para saber si estamos usando bien los modelos más complejos, si su score nos da peor al baseline probablemente se deba a un error de código.
- Nos sirve para rápidamente saber que tan complejo es un problema.
- Los modelos simples son fáciles de entender.

Requerimientos

- Implemente un perceptron
- Utilice todas las columnas del dataset (exceptuando columnas que no tenga sentido usar para predecir)
- Encoden las columnas que a necesarias para entrenar el modelo.
- Realice la busqueda de hiperparametros
- Responda:
  - ¿Cuál es el mejor score de validación obtenido? (¿Cómo conviene obtener el dataset para validar?)
  - Al predecir con este modelo para test, ¿Cúal es el score obtenido? (guardar el csv con predicciones para entregarlo después)
  - ¿Qué features son los más importantes para predecir con el mejor modelo? Graficar.

Validacion

Antes de proseguir con la parte III, deben validar con algun ayudante los resultados obtenidos en los items anteriores hasta el 27/10. El objetivo es poder darles una mano para interpretar los resultados obtenidos y que les sea menos complicada la siguiente etapa.

Parte III - Modelos (10 puntos)

Se deben entrenar (al menos) dos modelos **distintos**, incluyendo encodings y búsquedas de hiperparametros correspondientes. **Todos** los modelos a entregar deben cumplir:

- Utilizar F1-Score como métrica de **validación** (igual que la competencia).
- Deben medirse en **validación** y presentar el resultado (no es el calculado en la competencia).
- Deben ser reproducibles (Al correr el notebook varias veces, no debe afectarse el resultado).

Adicionalmente:

- Responder: ¿cómo conviene elegir los datos de validación respecto de los de train?.
- Responder: ¿Cuál es el mejor score en la competencia? (guardar el csv con predicciones para entregarlo después)
- En alguno de los modelos, se debe emplear como durante el feature engineering:
  - Imputación de nulos
  - Mean encoding en al menos una feature
  - One hot encoding en al menos una feature
- En alguno de los modelos se deben generar al menos 5 features nuevas (pueden venir vareas de la misma variable).

Puntos extra Kahoot

El alumnos que logren más podios en los Kahoot de ML (3 puntos por primer puesto, 2 puntos por segundo puesto, 1 punto por tercer puesto) suma 2 puntos extra en el TP; los siguientes 3 alumnos suman 1 punto extra.

Criterio de aprobación

Se necesita un 60% (12/20) de los puntos para aprobar, con al menos 4 puntos en los modelos.

Criterio de reentrega

Se podrá reentregar el TP si el puntaje es >=8, están todos los puntos desarrollados y tienen por lo menos 4 puntos en el punto de los modelos. La reentrega consiste en corregir todos los puntos donde tuvieran menos de la mitad de los puntos.

Se aprueba la reentrega si todos los puntos reentregados tienen al menos la mitad de los puntos. En caso de luego aprobar la instancia de reentrega, la nota es siempre 4.

Criterio de corrección

Parte I

- Cada visualización vale un punto

- Debe explicarse por sí misma, sin necesidad de texto aclaratorio.
- Debe tener rótulos en los ejes que corresponda y en el título (incluyendo unidades si corresponde).
- Debe mostrar una **relación con el target** que sea clara.
- El uso del color debe ser intencional, elegido por ustedes, no por la librería.
- La visualización debe ser legible (Un bar chart de 40 barras por ejemplo es ilegible)
- Debe cumplir el objetivo propuesto

Parte II

- Utiliza mal los datos de validación ya sea para obtener el resultado o para buscar hiper parámetros (-4 puntos). Ej:
  - El set de validación se usa para elegir los parámetros pero también está dentro del entrenamiento de cada modelo
  - El set de validación se usa filtrando información a los encodings
- El modelo no está bien hecho (-4 puntos), Ej:
  - Entrenan con las labels o datos cambiados para algunas filas
- No es capaz de predecir para la competencia o no lo hace correctamente (-4 puntos)
- No es reproducible (-2 puntos)
- No obtiene bien los features más importantes (-2 puntos)
- La predicción en la competencia da menos de 0.5 (-2 puntos)
- La predicción para la competencia tiene errores (-1 punto)
- No utiliza todos los features (-1 punto)

Parte III

- Para cada modelo cada condición no cumplida (o mal hecha) resta 1 punto.
- Feature engineering inapropiado para el modelo elegido (-2 puntos), Ej:
  - Features que no están normalizadas para una red neuronal.
  - Features sin ninguna consideración de escalas para un KNN.
- No buscan para todos los hiperparametros importantes (-2 puntos).
- Si un modelo diera un resultado menor a 0,6 en validación se invalida entero.

- Por sobre el puntaje total del ejercicio (ambos modelos) se restan 3 puntos si cualquiera de las siguientes cosas suceden (no acumulables):
  - Eligen mal el mejor modelo entre los dos
  - La predicción para la competencia no está bien hecha

    - La predicción en la competencia da menos de 0.5.

Si presentara más de 2 modelos, los puntos de los demás modelo se contarán como puntos extra (sobre la base de 2 puntos adicionales por modelo trabajado)

(A medida se acumulan estos pueden hacer que el item valga 0, pero nunca negativo)

Detalles y recomendaciones

- Para consultas conceptuales sobre machine learning o preguntas de consigna pueden consultar en el canal de slack #consultas-tp2.
- Para consultas de código con ayudante por privado. Recomendamos siempre consultar por disponibilidad en el canal publico.
- No deben buscar modelos entrenados por otros para usarlos, esto solo les puede jugar en contra porque es probable que no cumplan las condiciones pedidas, que no estén prolijos, que estén orientados a conseguir buenos resultados en la competencia y que tengan algún error conceptual.
- Recomendamos trabajar durante todo el TP en solo 4 notebooks: Uno de visualizaciones, otro para el baseline y uno para cada modelo de la parte III. Les recomendamos desarrollarlos de forma prolija y mostrar de forma ordenada cada uno de los resultados y pasos, con títulos y comentarios donde corresponda.
- El TP pide solo 6 visus y 3 modelos (baseline + 2 de competencia) con condiciones muy claras, tengan esa consideración a medida avanzan para chequear que cumplen todo.
- El TP **no pide ni evalúa más que lo que dice**, si bien ser original y tener un buen score suma en términos de trabajo y aprendizaje para ustedes, sean inteligentes respecto a los modelos y features que eligen para trabajar para garantizar que pueden terminar. Ya van a tener tiempo de ser originales en el TP3…
- Particularmente **este TP es muy difícil empezarlo al final**, en cuotas se vuelve mucho más sencillo. Sabemos que muchos de ustedes vienen haciendo algunos tps la última semana, y como les dijimos durante la presentacion, esperar que algo se entrene lleva inherentemente tiempo. La experiencia de cuatrimestres anteriores nos dice que **si no lo realizan de a poco no van a siquiera estar cerca de llegar**. Son demasiados conceptos a entender, muchas formas de hacerlo mal, y que sea una competencia le agrega un factor sorpresa respecto a problemas con solucion ya conocida. Esto no es solo una consigna a cumplir. El TP puede que sea más largo, pero se vuelve más corto mientras más temprano lo empiecen.
- Todos los puntos deben estar desarrollados.