Práctica 2 de Aprendizaje Estadístico, Curso 24/25

Master en Bioinformática, Universidad de Murcia
Juan A. Botía (juanbot@um.es), Gema M. Díaz (gemadiaz@um.es)
2024-11-28

1 Introducción

Esta es la segunda y última práctica de la asignatura de Aprendizaje Estadístico en el curso 2024/2025 correspondiente a los contenidos impartidos por el profesor Juan A. Botía.

Se pide, como resultado de la realización de la práctica, un R Markdown ejecutable (junto con el HTML correspondiente a la compilación de dicho Markdown) que haga las veces de memoria de resolución de la práctica y garantice la reproducibilidad de resultados.

El fichero R Markdown ha de ser suficientemente genérico para que el profesor pueda reproducir la totalidad de los análisis reportados por el alumno desde RStudio, usando Knit para generar un HTML. Cualquier análisis no documentado o no reproducible en el documento R Markdown a entregar, no se tendrá en cuenta a la hora de evaluar la práctica.

2 El conjunto de datos

El mecanismo de PCAP (Packet CAPture) es básicamente una API para capturar el tráfico de red (normalmente TCP/IP) para su posterior análisis en forma de paquetes de datos.

- Cada uno de esos paquetes capturados formará parte del flujo de datos de una conexión punto a punto entre dos direcciones IP.
- Dicha conexión se puede reconstruir y representar mediante un vector de características, a partir de obtener todos los paquetes de datos intercambiados en esa conexión.
- Los datos así obtenidos pueden usarse posteriormente para caracterización de tipos de conexiones de tal forma que podríamos diseñar modelos de machine learning para detectar qué tipo de conexión se acaba de producir o está produciéndose (e.g., una conexión fraudulenta o que supone un riesgo de seguridad).

2.1 El keylogger

Uno de esos riesgos de seguridad puede ser un keylogger. Este tipo de software actúa corriendo en segundo plano en nuestra máquina con un cometido muy simple: el de registrar todo lo que tecleamos. Un keylogger no tiene por qué ser necesariamente un software malicioso. Podemos usarlo, por ejemplo, como mecanismo para depuración de nuestras aplicaciones. Pero, como podemos imaginar, puede usarse fraudulentamente para, entre otros usos, capturar nuestras credenciales al entrar en nuestras cuentas en Internet.

Vamos a trabajar con un conjunto de estas características, obtenido de Kaggle.

El fichero sobre el que vamos a trabajar se ha de descargar del área de recursos del Aula Virtual, disponible aquí. Se ha de bajar, y descomprimir. Téngase en cuenta que ocupa más de 300MB por lo que ha de manejarse con cuidado si nuestra máquina no va sobrada de recursos. Por ejemplo, puedes usar este fragmento de código como ejemplo para evitar prototipar tu notebook con todos los datos generando una muestra del 10%

```
awk 'BEGIN {srand()} NR==1 {print > "sampled.csv"; next} rand() <= 0.1 {print >> "sampled.csv"}' Keylogge
```

Una vez abierto en RStudio, podemos ver que contiene más de 90K ejemplos, y un total de 86 características que describen, en cada una de sus filas, una conexión entre dos direcciones IP.

- Características sobre el flujo: Flow.ID, Source.IP, Source.Port, Destination.IP, Destination.Port, Protocol, Timestamp
- Características sobre los paquetes del flujo: Flow.Duration, Total.Fwd.Packets, Total.Backward.Packets, Total.Length.of.Fwd.Packets, Total.Length.of.Bwd.Packets,
    - Fwd.Packet.Length.Max, Fwd.Packet.Length.Min, Fwd.Packet.Length.Mean, Fwd.Packet.Length.Std,
    - Bwd.Packet.Length.Max, Bwd.Packet.Length.Min, Bwd.Packet.Length.Mean, Bwd.Packet.Length.Std,
    - Flow.Bytes.s, Flow.Packets.s
- Características sobre el tiempo transcurrido entre un paquete y el siguiente: ç
    - Flow.IAT.Mean, Flow.IAT.Std, Flow.IAT.Max, Flow.IAT.Min,
    - Fwd.IAT.Total, Fwd.IAT.Mean, Fwd.IAT.Std, Fwd.IAT.Max, Fwd.IAT.Min
    - Bwd.IAT.Total, Bwd.IAT.Mean, Bwd.IAT.Std, Bwd.IAT.Max, Bwd.IAT.Min

Y algunas otras. Todas pueden encontrarse listadas aquí

3 La tarea

Vamos a realizar análisis orientados a producir un modelo de clasificación lo más preciso posible. Por tanto, la práctica se reduce a

- realizar un análisis basado en técnicas estadísticas y de machine learning vistas en clase para obtener un modelo que clasifique de manera aceptable los ejemplos del conjunto de test y
- generar un informe que incluya el propio proceso de análisis y la documentación relativa a cómo se ha realizado.
- **Se han de trabajar al menos dos algoritmos de entre los vistos en clase, concretamente: bagging, boosting, svm y redes neuronales**.

Una vez obtenido el fichero Keylogger tal cual viene del csv, se debe reservar un conjunto de test aparte, que obtendremos mediante la función `sample()`. Justo antes de invocar a sample, usaremos la función `set.seed()` con la semilla 12345. Por ejemplo, como en

```
set.seed(12345)
test = mydf[sample(1:nrow(mydf),10000),]
```

Este conjunto de test se usará para obtener un estimador puntual del error en las distintas estrategias que generemos.

En la resolución de la práctica se han de demostrar las siguientes destrezas:

- Capacidad para entender el funcionamiento interno de dos o más algoritmos de aprendizaje pertenencientes a diferentes paradigmas, sus hiperparámetros y los rangos de valores que, a priori, dichos parámetros podrían tomar
- Capacidad para optimizar hiperparámetros de un algoritmo de machine learning.
- Capacidad para diseñar y ejecutar estrategias para reducción de la dimensionalidad de conjuntos de datos.
- Capacidad para visualizar datos y resultados.
- Capacidad crítica avanzada para analizar resultados y extraer conclusiones.
- Destreza en la producción de análisis, documentados y reproducibles mediante RStudio y R Markdown.

4 El proceso de la selección y validación de modelos

Definimos la selección de modelos como el proceso por el cual, dado un conjunto de datos D y un algoritmo de aprendizaje estadístico, se llevan a cabo los siguientes pasos, para una validación cruzada de n pliegues

1. Dividimos los datos entre conjunto de train y test
2. Dividimos los datos de training en $k = n$ folds (pliegues)
3. Para cada algoritmo concreto a, de entre los dos/tres que se han de utilizar

    3.1 Definimos un conjunto de valores posibles a probar, para los hiperparámetros $v_1, v_2, \ldots, v_j$ para a. Cada v_i es un vector de m valores, uno para cada uno de los m hiperparámetros de a

    3.2 Para cada v_i, $1 \leq i \leq j$

    - Para cada fold k
        - Entrenar el modelo con todos los folds excepto k
        - Obtener un error de test del modelo con el fold k
    - Calcular media y desviación estándar para el error de validación cruzada

    3.3 se selecciona el conjunto de valores v^* para los hiperparámetros de entre los v_i, $1 \leq i \leq j$ con mejor *accuracy* según la regla *one-standard-error*

    3.4 Entrenamos un nuevo modelo con a, en donde sus hiperparáemtros toman los valores v^* y usamos todos los ejemplos del cjto. de train

    3.5 Estimamos el error de dicho modelo según el conjunto de test

5 Trabajo a realizar

La tarea que han de resolver los modelos que produzcamos es la de predecir si la conexión es fraudulenta (la columna `class` del conjunto de datos).

Para ello, se han de responder a las siguientes preguntas

1. [1/10] Agrupar todas las variables del conjunto por temática, describiendo lo que significa cada uno de los grupos de características que has identificado. Se pide básicamente, terminar el trabajo empezado arriba cuando se han descrito las características relativas al flujo, a los paquetes del flujo, y tiempo transcurrido entre paquete y paquete. Identifica y describe las ya identificadas arriba junto con las que faltan. Dentro de las características de cada conexión, para la mayoría de categorías de características, se distingue entre características forward y backward. Explicar en que se diferencian unas de otras y mencionar las referencias documentales que se han usado para responder. Desde el punto de vista del análisis de datos, ¿es relevante diferencia entre características forward y backward? ¿Por qué?
2. [0.5/10] Plantear y ejecutar una estrategia que te permita trabajar con el conjunto de datos de manera holgada si tener que usar necesariamente los 90K ejemplos en memoria.
3. [0.5/10] Elabora un análisis no supervisado (PCA) de las conexiones y explica lo que ves. ¿Cómo dirías que va a ser el problema de la elaboración de un modelo de clasificación que identifique conexiones de keystrokes en términos de dificultad? Razona la respuesta.
4. [2/10] ¿Es posible identificar si la conexión es benigna o maligna mediante un modelo de regresión? ¿Qué tipo de modelo de regresión habría que usar en este caso? ¿Cuáles son las características relevantes? ¿Cómo se comporta el modelo?
5. [5/10] Para cada uno de los dos algoritmos escogidos
    - Explicar brevemente el tipo de modelo que genera el algoritmo, y cuál es la estrategia de dicho algoritmos para construir el modelo.
    - Indicar si el algoritmo en cuestión tiene algún requisito en cuanto a si se han de preprocesar los datos (e.g. escalado, imputación de valores nulos, etc.) y cómo.
    - Identificar y explicar cada uno de sus hiperparámetros
    - Detallar una estrategia para la generación del *grid* de valores para hiperparámetros a usar.
    - Ejecutar la estrategia, generar los modelos y seleccionar el mejor siguiendo la estrategia explicada arriba
6. [1/10] Responde a la pregunta de qué algoritmo de ML ha funcionado mejor para este problema y por qué, analizando sus parámetros y los resultados

6 Sobre el material a entregar

- Se ha de entregar un fichero Markdown fuente junto con el compilado.
- El R Markdown fuente ha de poder ejecutarse en su totalidad por el profesor, desde RStudio mediante Knit.
- La única entrada que se espera usar es el fichero `csv` mencionado arriba.
- No se tendrá en consideración ningún análisis que, aunque se mencione en el informe, no se incluya como tal mediante su código fuente y la generación automática de los resultados.
- Los ficheros RDS se harán llegar al profesor, junto con el Markdown fuente y el compilado a través de la herramienta `Tareas` del aula virtual. Téngase en cuenta que el sistema Tareas del AV tiene un límite de 100MB. Si vuestros ficheros R Markdown fuente + R Markdown compilado + ficheros RDS no superan ese límite, por favor, haced una entrega completa por ese sistema. Si la entrega supera ese límite, podéis usar el servicio FileSender. Dicha consigna os proporcionará un link, que debéis incluir en el texto de la entrega y que yo usaré para descargarme el zip con los ficheros necesarios.

7 Algunos consejos a tener en cuenta

- Si el documento Markdown os resulta muy largo, se aconseja dividir el desarrollo en documentos Markdown más pequeños y una vez estén listos todos los análisis, se puede ensamblar todo en el documento final.
- Seguramente tendremos chunks de código llamadas a funciones que tardan demasiado y que resultan un fastidio si necesitamos ejecutarlas repetidas veces cuando estamos a medias de desarrollar algo. Se aconseja, para que la resolución de la práctica sea más ágil, los siguientes pasos: (1) la variable que almacena el resultado de dicha función se ha de guardar en un fichero, con `saveRDS`, (2) el chunk de código se anula con `eval=FALSE` en el prólogo del chunk para que no se evalue más, (3) el siguiente chunk de código hace uso del resultado leyéndolo del fichero con `readRDS`.
- Para una entrega de la práctica sin problemas se aconseja, antes de enviarla, asegurarse de que compila de principio a fin.

Nótese que para la resolución de la práctica es más que suficiente con los paquetes R que se indican arriba, y que se estudian durante la segunda parte de la asignatura. En todo caso, si el alumno se ayuda de algún paquete adicional que redunde en una mejor calidad en los análisis, será bienvenido y valorado positivamente.

8 Fecha de entrega

La fecha límite para entrega de la práctica será el 20 de Diciembre de 2024, según acuerdo de la comisión académica del máster. Se abrirá la correspondiente tarea para ello.