Tarea programada 2

De barcos, árboles, bosques, y alitas Red Bull

Usted ha sido enviado al pasado. Específicamente, al 15 de abril de 1912, en una fría noche en el océano Atlántico. Los pasajeros del barco aún no lo saben, pero usted, que viene del futuro, si: está a punto de ocurrir un accidente que cobrará la vida de 1502 personas. Con usted, viajan 2000 pastillas Red Bull, que dan alas y que permitirán salvar la vida de esas 1502 personas que, sin la tecnología actual, habrían enfrentado un destino diferente. Ha llegado la hora de salvar a los pasajeros del Titanic, ¡que comience la misión!

Parte I - La historia

Estamos en el año 2525. La capacidad de viajar al pasado ha sido ampliamente estudiada, por lo que ahora es posible que un estudiante de la U cumpla con sus horas de TCU trabajando en alguna época pasada. Usted ha decidido salvar a los pasajeros del Titanic. En materia de viajes al pasado, está prohibido por ley saber el nombre de las personas con las que tratamos, para no afectar el futuro de la humanidad directamente. Por lo tanto, usted ha decidido apoyarse en lo que llaman un bosque aleatorio (*random forest*), un algoritmo de aprendizaje automático supervisado, que procura clasificar instancias de un problema en dos o más categorías. Su misión consiste entonces en construir una herramienta que adivine lo mejor posible si un(a) pasajero(a) necesita alas Red Bull, con base en su edad, sexo y otras características.

Parte II - Árboles de decisión y bosque aleatorio

Los árboles de decisión son modelos clasificadores que pertenecen a la categoría de aprendizaje automático supervisado. Estos algoritmos se entrenan conociendo de antemano cuál es la salida esperada para cada instancia que se debe clasificar. En el problema específico del Titanic, usted recibe un archivo que consta de las siguientes columnas: sobrevivió (0 o 1), sexo (femenino o masculino), clase (1, 2 o 3 para referirse a primera, segunda y tercera clase), así como cantidad de familiares en el barco (entero). La columna “sobrevivió” es la etiqueta, se sabe de antemano quiénes sobrevivieron y quiénes no. El objetivo del árbol de decisión, y de cualquier algoritmo de aprendizaje supervisado en general, es identificar patrones en el resto de los datos (sexo, clase y cantidad de familiares) para tratar de predecir si una persona sobrevivió o no.

Cuadro 1. Ejemplo ficticio de datos del Titanic

Fila	Sobrevivió	Sexo	Clase	Familiares
1	0	Masculino	1	3
2	1	Femenino	2	4
3	1	Femenino	1	4

4	1	Femenino	2	4

Observe el cuadro 1 y trate de construir su propia clasificación. Algunos clasificadores funcionan bien para este ejemplo reducido, por ejemplo: “si el sexo es femenino, la persona sobrevive”, con 100% de precisión. El clasificador “las personas con 4 familiares sobreviven” también obtiene un 100% de precisión. Sin embargo, el clasificador “Las personas de clase 2 sobreviven” únicamente obtiene un 50% de precisión. En cualquiera de los casos, note como la forma mental de construir el clasificador es buscar qué patrones garantizan la mayor efectividad para predecir el resultado final.

Los árboles de decisión intentan detectar estos mismos patrones y generar una estructura de datos que permite ir haciendo preguntas sobre cada instancia que se debe clasificar, para decidir si pertenece a una clase o la otra. La figura 1 muestra un árbol de decisión cuyo razonamiento es preguntar por el sexo de la persona. Observe como hay dos tipos de nodos: de clasificación o de decisión. En este caso, el nodo raíz es un nodo de clasificación, que hace una pregunta. Con base en la respuesta, toma un camino diferente en el árbol. En este caso en particular, los nodos hoja son nodos de decisión.

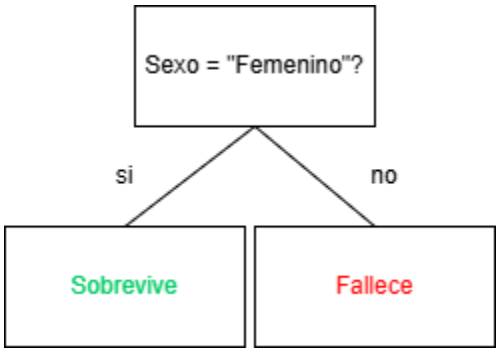

Figura 1. Ejemplo de árbol de decisión

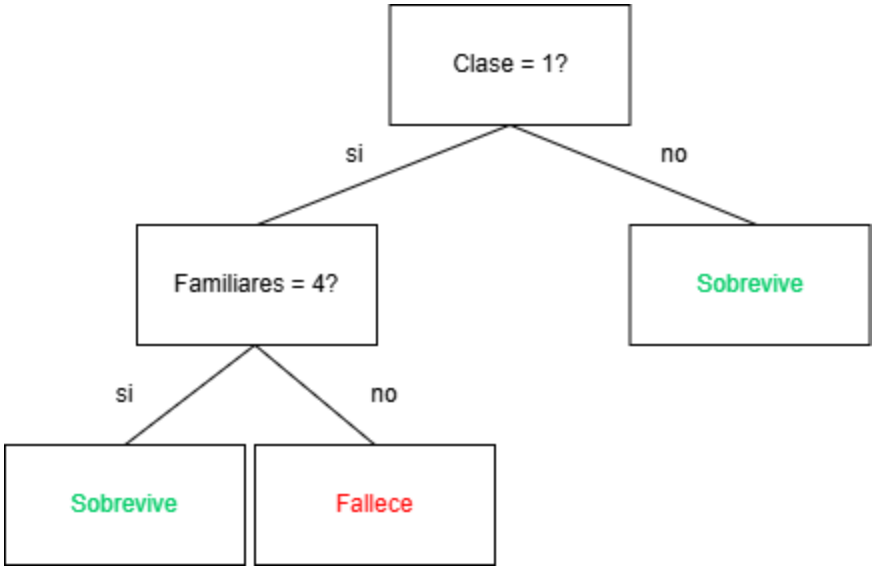

Figura 2. Otro ejemplo de árbol de decisión

La figura 2 muestra otro árbol que también clasifica adecuadamente todas las instancias. Sin embargo, note que esta vez se hacen más preguntas. Esto es innecesario, ya que el árbol de la figura 1 es claramente más eficiente. Entre menos preguntas se deban hacer, mejor. Además, en ocasiones no es posible crear el árbol perfecto, capaz de clasificar correctamente todas las instancias.

Los bosques aleatorios son un conjunto de árboles de decisión, en donde la decisión final para clasificar una instancia se hace a través de una votación entre todos los árboles. Este sistema permite subsanar decisiones equivocadas cuando alguno de los árboles no fue construido de la mejor manera. Volviendo al ejemplo, los árboles de las figuras 1 y 2 formarían un bosque aleatorio de 2 elementos.

Parte III - Los requerimientos

A continuación se listan los requerimientos de esta tarea

1) El archivo CSV con el que se le ha permitido trabajar es una versión reducida del archivo disponible aquí. En este dataset, cada fila es una **Instancia**, que consta de los siguientes atributos, en orden: sobrevivió (0 o 1), sexo (femenino o masculino), clase (1, 2 o 3 para referirse a primera, segunda y tercera clase), así como cantidad de hermanos/pareja en el barco (entero) y la cantidad de padres/hijos en el barco (entero).
2) Una de sus primeras responsabilidades consiste en leer este archivo línea por línea, para crear dos **Dataset** (entrenamiento y evaluación), que constan cada uno de una **lista** de objetos **Instancia**. Para crear estos dos conjuntos, debe asignar cada línea nueva al dataset de entrenamiento, con una probabilidad de 80% o bien al de evaluación, con una probabilidad del 20%.

3) Una vez que tenga una representación de los Dataset en la computadora, deberá crear un **bosque aleatorio con *n* árboles binarios**, donde *n* será solicitado al usuario. En cada árbol, el criterio para detenerse es que el árbol tenga una profundidad *p*, que se solicita al usuario antes de comenzar. Los *n* árboles del bosque aleatorio deben crearse usando backtracking y fuerza bruta. Cada estudiante debe elegir con qué criterios crea los árboles, pero se asignarán puntos extra a ciertos/as estudiantes que consigan una mejor precisión en sus modelos y una mejor explicación de cómo crearon los árboles de una manera más 'inteligente'.
4) Con el bosque creado, su programa debe ofrecer al usuario un menú con las siguientes opciones:
    a) Mostrar el *i*-ésimo árbol, donde *i* se solicita al usuario. Su programa debe mostrar algo que permita reconocer la estructura del árbol y, sobre todo, cuál es la pregunta que se hace en cada nodo de decisión.
    b) Clasificar una instancia nueva, cuyos datos se solicitan al usuario (el programa deberá indicar la decisión de cada árbol y el resultado de la votación)
    c) Ejecutar el proceso completo de evaluación, que significa clasificar todas las instancias del Dataset de evaluación y reportando en un archivo de salida clasificacion.txt:
        i) Al inicio del archivo, las siguientes cuatro relaciones:
            - TP: Cantidad de personas clasificadas como sobrevivientes / Cantidad total de personas sobrevivientes según el dataset
            - FP: Cantidad de personas clasificadas como no sobrevivientes / Cantidad total de personas sobrevivientes según el dataset
            - FN: Cantidad de personas clasificadas como no sobrevivientes / Cantidad total de personas no sobrevivientes según el dataset
            - TN: Cantidad de personas clasificadas como no sobrevivientes / Cantidad total de personas no sobrevivientes según el dataset
        ii) Por cada instancia: el número de instancia, la clasificación de cada árbol, la clasificación final del bosque aleatorio
5) Entrega: cada persona deberá trabajar en el repositorio de código que se le asignará. Se debe entregar un archivo .jar para ejecutar la solución y este ejecutable debe permitir agregar el nombre del archivo CSV desde la línea de comandos. En el repositorio también podrá existir un archivo explicacion.txt donde cada estudiante explica qué criterios usó para crear mejores árboles. Si este archivo no existe, se asumirá que la persona estudiante hizo fuerza bruta aleatoria, lo cual es válido y no será penalizado.
6) Fecha de entrega: Viernes 21 de noviembre