Cancer de mama

Machine Learning. PEC 3. 2020

Introducción

El cribado del cáncer de mama permite diagnosticar y tratar la enfermedad antes de causar síntomas notables. El proceso de detección temprana implica examinar el tejido mamario en busca de bultos o masas anormales. Si se encuentra un bulto, se realiza una biopsia por aspiración con aguja fina, que utiliza una aguja hueca para extraer una pequeña porción de células de la masa. Luego, un médico examina las células bajo un microscopio para determinar si es probable que la masa sea maligna o benigna.

En este PEC, investigaremos la utilidad del aprendizaje automático para detectar el cáncer en las mediciones de células biopsiadas de mujeres con masas mamarias anormales.

Utilizaremos datos de cáncer de mama que incluye mediciones de imágenes digitalizadas de aspiración con aguja fina de una masa mamaria. Los valores representan características de los núcleos celulares presentes en la imagen digital.

En el fichero BreastCancer1.csv estan los datos sobre el cáncer de mama de 569 casos de biopsias de cáncer, cada uno con 32 características. La primera característica es un número de identificación, después son 30 mediciones de laboratorio con valores numéricos y por último, esta el diagnóstico. El diagnóstico se codifica como `M` para indicar maligno o `B` para indicar benigno.

Objetivo:

En esta PEC se analizan esto datos mediante la **implementación** de los diferentes **algoritmos estudiados**: *k-Nearest Neighbour, Naive Bayes, Artificial Neural Network, Support Vector Machine, Arbol de Decisión y Random Forest* para **diagnosticar** el tipo de cáncer de mama.

Puntos importantes:

1. Realizar una exploración de los datos que incluya una estadística descriptiva básica de las variables mediante tablas y gráficos.
2. En cada algoritmo hay que realizar las siguientes tres etapas: 1) Transformacion de los datos (en caso necesario) 2) Entrenar el modelo 3) Predicción y Evaluación del algoritmo. En la fase 3) "tunear" diferentes valores de los hiperparámetros del algoritmo para posteriormente evaluar su rendimiento.
3. Se debe aplicar la misma selección de datos training y test en todos los algoritmos. Utilizando la semilla aleatoria `12345`, para separar los datos en dos partes, una parte para training (67%) y otra parte para test (33%). Si se prefiere, se puede escoger otro tipo de partición de los datos para hacer la selección de training y test como por ejemplo k-fold crossvalidation, boostrap, random splitting, etc. Lo que es importante es mantener la misma selección para todos los algoritmos.
4. En todos los casos se evalua la calidad del algoritmo con la información obtenida de la función `confusionMatrix()` del paquete `caret`.
5. Para la ejecución especifica de cada algoritmo se puede usar la función de cada algoritmo como se presenta en el libro de referencia o usar el paquete `caret` con los diferentes modelos de los algoritmos. O incluso, hacer una versión mixta.

6. Comentario sobre el informe dinámico. Una opción interesante del knitr es poner `cache=TRUE`. Por ejemplo:

   ```
   knitr::opts_chunk$set(echo = FALSE, comment = NULL, cache = TRUE)
   ```

   Con esta opción al ejecutar el informe dinámico crea unas carpetas donde se guardan los resultados de los procesos. Cuando se vuelve a ejecutar de nuevo el informe dinámico solo ejecuta codigo R donde se ha producido cambios, en el resto lee la información previamente descargada. Es una opción muy adecuada cuando la ejecución es muy costosa computacionalmente.

Informe de la PEC

Las soluciones se presentarán mediante un informe dinámico R markdown con la siguiente estructura:

1. Titulo: igual que el de la PEC, autor, fecha de creación e indice de apartados de la PEC.
2. Sección de lectura, exploración de los datos y obtención de los muestras de train y test. Recordar que un primer paso es, si hace falta, transformar las variables leidas al tipo de objeto R adecuado al tipo de variable. La exploración de los datos se aplica a todas las variables leidas. (*Puntuación*: 10%)
3. Sección de aplicación de cada algoritmo para la clasificación. Está formado por subsecciones que corresponden a cada algoritmo: k-Nearest Neighbour, Naive Bayes, Artificial Neural Network, Support Vector Machine, Arbol de Decisión y Random Forest manteniendo este orden. (*Puntuación*: 60%)

   En cada algoritmo hay que realizar las tres etapas mencionadas anteriormente.
4. Sección de conclusión y discusión sobre el rendimiento, interpretabilidad, ... de los algoritmos para el problema tratado. Proponer que modelo o modelos son los mejores. (*Puntuación*: 20%)

Un característica que se valorará es hasta que punto es el informe "dinámico". En el sentido de adaptarse el informe a cambios en los datos, es decir, si el fichero de datos cambia el informe se adapta a los nuevos resultados. (*Puntuación*: 10%)

Se subiran al registro de entregas un **zip** con los siguientes ficheros:

1. Fichero ejecutable (.Rmd) que incluya un texto explicativo que detalle los pasos implementados en el script y el código de los análisis. No olvidar de incluir todos los ficheros complementarios que hagan falta para la correcta ejecución: *ficheros de datos, fichero de bibliografia, imagenes, ...*

   NOTA: Para facilitar la ejecución, no usar un ruta fija para la lectura del fichero, asociarlo al area de trabajo donde este el fichero .Rmd.
2. Informe (pdf) resultado de la ejecución del fichero Rmd anterior.

Antes de enviar el zip, se recomienda **verificar la reproducibilidad del fichero .Rmd** para obtener el informe en formato pdf sin ninguna dificultad.