uc3m | Universidad Carlos III de Madrid

OPENCOURSEWARE

APRENDIZAJE AUTOMÁTICO PARA EL ANÁLISIS DE DATOS

GRADO EN ESTADÍSTICA Y EMPRESA

Ricardo Aler

PRÁCTICA DE REGRESIÓN CON MLR. PREDICCIÓN DEL PRECIO DE LA VIVIENDA.

(3 puntos)

INTRODUCCIÓN

El objetivo de la práctica es aplicar técnicas de aprendizaje automático de regresión, para predecir el precio de la vivienda basándose en atributos tales como la zona de la ciudad, los metros que comparte con la calle, el tipo de calle, la pendiente del terreno, la cercanía al transporte público o los materiales de construcción. En total usa 79 atributos de entrada, cuyo significado se puede consultar en:

https://www.kaggle.com/c/house-prices-advanced-regression-techniques/data

Esta práctica está basada en la competición de Kaggle "House Prices / The Ames Housing dataset" (https://www.kaggle.com/c/house-prices-advanced-regression-techniques).

Evaluación de la práctica:

- Cada apartado tiene su puntuación
- Todos los apartados son obligatorios, excepto el 4 (ensembles) y el 5 (feature selection), que son opcionales (a costa de perder la puntuación correspondiente, claro está).
- La valoración de cada apartado dependerá del código que se escriba, de los resultados obtenidos, pero también de cómo se describa en la memoria tanto lo realizado como los resultados. En apartados donde se requiera una comparación de resultados, se recomienda utilizar tablas.
- Es necesario entregar el código realizado, pero también una memoria donde se describa el código escrito y un resumen de los resultados, con tablas comparativas.

---

1) (0.3 puntos) Carga y transformación inicial de los datos: los datos están en el fichero "train_precios.csv".

- Cargadlos en memoria con *read.csv*.
- Haced una exploración inicial de los datos:
  - **i.** ¿Cuántas instancias y cuántos atributos hay?
  - **ii.** Usad *summary* para haceros una idea de los datos.
  - **iii.** Usad *sapply* para saber:
    1. Cuántos atributos (excluyendo la variable de respuesta) hay de cada tipo (factores, numéricos, enteros, character, …)
    2. En qué columnas de los datos hay *NA's*, qué proporción de NA's hay en ellas (= número de datos con NA's en relación al número total de datos), y ordenad las columnas por proporción de NA's. ¿Es preocupante esta proporción en algunos atributos?
- Haced un primer pre-proceso de los datos:
  - **i.** ¿Tiene sentido la columna *Id*? Considerad eliminarla.
  - **ii.** El atributo a predecir es ***SalePrice*** (precio de venta). Sin embargo, las normas de la competición dicen que se va a usar **RMSE** como medida de evaluación, pero del logaritmo de *SalePrice* (para que los pisos de precio muy alto no tengan un peso desproporcionado frente a los de precio bajo). Así, transformaremos los datos para que la columna a predecir sea *log(SalePrice)* transformando la columna *SalePrice* en *log(SalePrice)*.

2) **(0.6 puntos) Exploración inicial de modelos:** Vamos a probar varios métodos para hacernos una idea de cuáles funcionan mejor y cuánto tardan (coste computacional).

- Probaremos varios tipos: árboles de regresión (*rpart*), árboles de modelos (*cubist*), *knn* para regresión, un modelo lineal normal (lm), y support vector machine para regresión con dos kernels, uno *linear*, el otro *rbf* (*svm*),.
- Tenéis que averiguar qué nombre tiene cada uno de esos métodos en MLR, buscando en esta página. https://mlr-org.github.io/mlr/articles/tutorial/integrated_learners.html#regression-60

- Para métodos como *svm* y *knn*, en principio habría que normalizar las entradas. Afortunadamente, estos dos métodos concretos ya hacen una estandarización interna, con lo que no tenemos que hacerlo nosotros explícitamente.
- Ocurre que algunos métodos son capaces de manejar datos con *NA's* y otros no, y algunos métodos son capaces de manejar atributos categóricos y otros no (factores). Es necesario comprobar en la página qué propiedades tiene cada método (columnas NAs y fac., entre otras).
- Como algunos métodos de aprendizaje son capaces de manejar *NA's* directamente, pero otros no, definiremos dos tareas de regresión: una con los datos originales (sin imputar), otra con los datos imputados (con la media y la moda), y tal vez otra con datos imputados y con variables dummy (con *createDummyFeatures*), si algún método lo requiere.
- Una vez definidas las tareas, definid el método de evaluación (*makeResampleDesc* y *makeResampleInstance*) con validación cruzada de 5 folds.
- Después, calculad los errores usando *resample* (mirad el tutorial)
- Finalmente, mostrad los errores cometidos por cada método (por ejemplo, *errores_lm$aggr*) y el tiempo que tarda cada método (*errores_lm$runtime*)
- Por tanto, para cada método, habréis obtenido su error y su tiempo. Extraed algunas conclusiones: ¿cuál es el mejor método de momento (sin ajustar hiper-parámetros)? ¿Merecen la pena los modelos no lineales frente a los lineales? ¿El mejor método lo es a costa de ser muy lento?
- Para el método que mejor funcione, comprobar si imputar con la mediana proporciona mejores resultados.

3) **(0.6 puntos) Ajuste automático de hiper-parámetros.** Usad dos métodos de ajuste:
   - RandomSearch
   - Model-based optimization: paquete mlrMBO (tutorial en mlr https://mlr-org.github.io/mlrMBO/articles/supplementary/machine_learning_with_mlrmbo.html)

Tras hacer la evaluación siguiendo el mismo método que en la sección anterior (2), es decir, validación cruzada con 5 folds, ¿cuál de los dos métodos obtiene mejores resultados? ¿Cuál tarda menos? ¿Se consigue mejorar los resultados con respecto a los que obtuvisteis en el apartado 1, antes de ajustar hiper-parámetros?

4) **(0.1 puntos) Competición KAGGLE:** Con el mejor modelo obtenido (y los mejores hiper-parámetros), enviad los resultados a la competición (Submit Predictions en *https://www.kaggle.com/c/house-prices-advanced-regression-techniques*). Para ello hay que registrarse en Kaggle con un nombre de equipo y enviar las salidas de vuestro mejor modelo ("make a submission") sobre el conjunto de test que os he dejado en Áula Global. En esta sección tenéis que documentar el nombre de equipo que habéis usado, cuál es el error que os aparece en el Dashboard y en qué orden quedasteis. **Tenéis que proporcionar un volcado de pantalla en la memoria del envío a la competición.** Importante: recordad que todos los modelos que habéis construido predicen *log(SalePrice)*, pero la competición pide directamente *SalePrice*, por lo que antes de enviar los resultados a la competición, tendréis que hacer esta transformación: $SalePrice = 10^{precio}$.

5) (0.6 puntos) Construid modelos avanzados con Random Forests, Extremely Randomized Trees y Gradient Tree Boosting, todos con ajuste automático de hiper-parámetros. Igual que en secciones anteriores, hay que realizar la evaluación con validación cruzada con 5 folds.

6) (0.5 puntos) Construid una secuencia de métodos para hacer *feature selection* mediante algún método *filter*. Hacedlo de tal manera que el número de atributos a seleccionar sea un hiper-parámetro de la secuencia de métodos, de tal manera que se pueda ajustar también mediante ajuste automático de hiper-parámetros. Igual que en secciones anteriores, hay que realizar la evaluación con validación cruzada con 5 folds.

7) **(0.3 puntos) Competición KAGGLE:** Volved a enviar los resultados del mejor modelo a Kaggle. Documentad el envío en la memoria