Reto Real Estate Modelling

Ramon Ruiz-Dolz

Grupo de Tecnología Informática - Inteligencia Artificial

April 16, 2019

Índice

Índice

Dominio del Problema

- Competición anual de *Data Science* impulsada por Cajamar.
- Datos reales extraídos del portal HAYA real state
- *Dataset* de entrenamiento compuesto por 9958 muestras (viviendas)
- Cada vivienda cuenta con 52 características más el *target*
- Características agrupadas en 3 subgrupos (HY, IDEA, GA):
  - HY: Datos específicos de la vivienda (e.g. num. habitaciones, baños, tipo de vivienda, etc.)
  - IDEA: Datos estadísticos de la vivienda (e.g. densidad de población de la zona, % de uso residencial, oficinas, etc.)
  - GA: Datos relacionados con el trafico web de la página

Objetivos

- Analizar y procesar la información proporcionada
- Hallar las características de mayor importancia a partir del conjunto de datos disponible
- Predecir la duración media de una visita a una página web.
- Minimizar la métrica *"Median Absolute Error Loss"*[1] entre el valor objetivo (*target*) y la predicción

[1]

$$MAE = median(|y - x|) \tag{1}$$

Índice

Análisis de los Datos

- Jupyter Notebook con Python 3
- Análisis individualizado para cada característica (i.e. cantidad de valores diferentes, observación de outliers, distribución de los valores, etc.)
- Inferencia de características incompletas (e.g. província a partir de cod. postal)
- Búsqueda de máximos, mínimos, valores desconocidos, etc.

Procesado de Características (I)

- Codificación One-hot: Se ha aplicado este tipo de codificación a las variables categóricas (e.g. HY_provincia, HY_tipo, HY_cert_energ, etc.)

- Transformación a valores booleanos: Se ha aplicado dicha transformación a variables textuales (e.g. HY_descripcion, HY_distribucion, etc.)

- Combinación de variables: Se han combinado variables con el fin de completar valores perdidos y extraer nuevo conocimiento del dataset (e.g. HY_metros_totales a partir de HY_descripcion y HY_distribucion con RegEx, GA_page_views con GA_bounce_rate y GA_exit_rate partiendo de sus definiciones[2])

[2] `https://conversionxl.com/guides/bounce-rate/bounce-rate-vs-exit-rate/`

Procesado de Características (II)

- Predicción de valores perdidos: Se han entrenado otros modelos con suficiente precisión como para completar algunos de los valores inexistentes (e.g. HY_antiguedad)
- Completar valores perdidos: Se probó a completar los valores perdidos tanto con la media como a ceros
- Eliminación de outliers: Se eliminaron los outliers detectados en la fase previa de análisis

Índice

Gestión de los Datos

- Tamaño del *dataset* preprocesado: 8166
- Partición del *dataset* mediante la técnica de 10-Fold con barajado
- Tamaño de train en cada fold: 7349
- Tamaño de dev en cada fold: 817
- Tamaño de test: 1103

Modelos Considerados

- K-vecinos (K-NN)
- Support Vector Regressor (SVR)
- Redes Neuronales (NN)
- Random Forests
- **Gradient Tree Boosting**
- Gradient Boosting Regressor (XGBoost)

Gradient Tree Boosting

1. Inicialización del modelo con un valor constante:

$$F_0(x) = \arg\min_{\gamma} \sum_{i=1}^{n} L(y_i, \gamma) \tag{2}$$

2. En cada iteración de m = 1 a M:
   1. Calcular los *pseudo-residuals*:

$$r_{im} = -[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}]_{F(x)=F_{m-1}(x)} i = 1, \ldots, n \tag{3}$$

   2. Ajustar un árbol de decisión a los *pseudo-residuals*
   3. Hallar el multiplicador γ_m:

$$\gamma_m = \arg\min_{\gamma} \sum_{x_i \in R_{jm}} L(y_i, F_{m-1}(x_i) + \gamma) \tag{4}$$

   4. Actualizar el modelo:

$$F_m(x) = F_{m-1}(x) + \sum_{j=1}^{J_m} \gamma_{jm} \mathbf{1}_{R_{jm}}(x) \tag{5}$$

3. Salida de $F_M(x)$

Resultados

- Resultado en entrenamiento: 16.44 MAE
- Resultado en test: 18.92 MAE

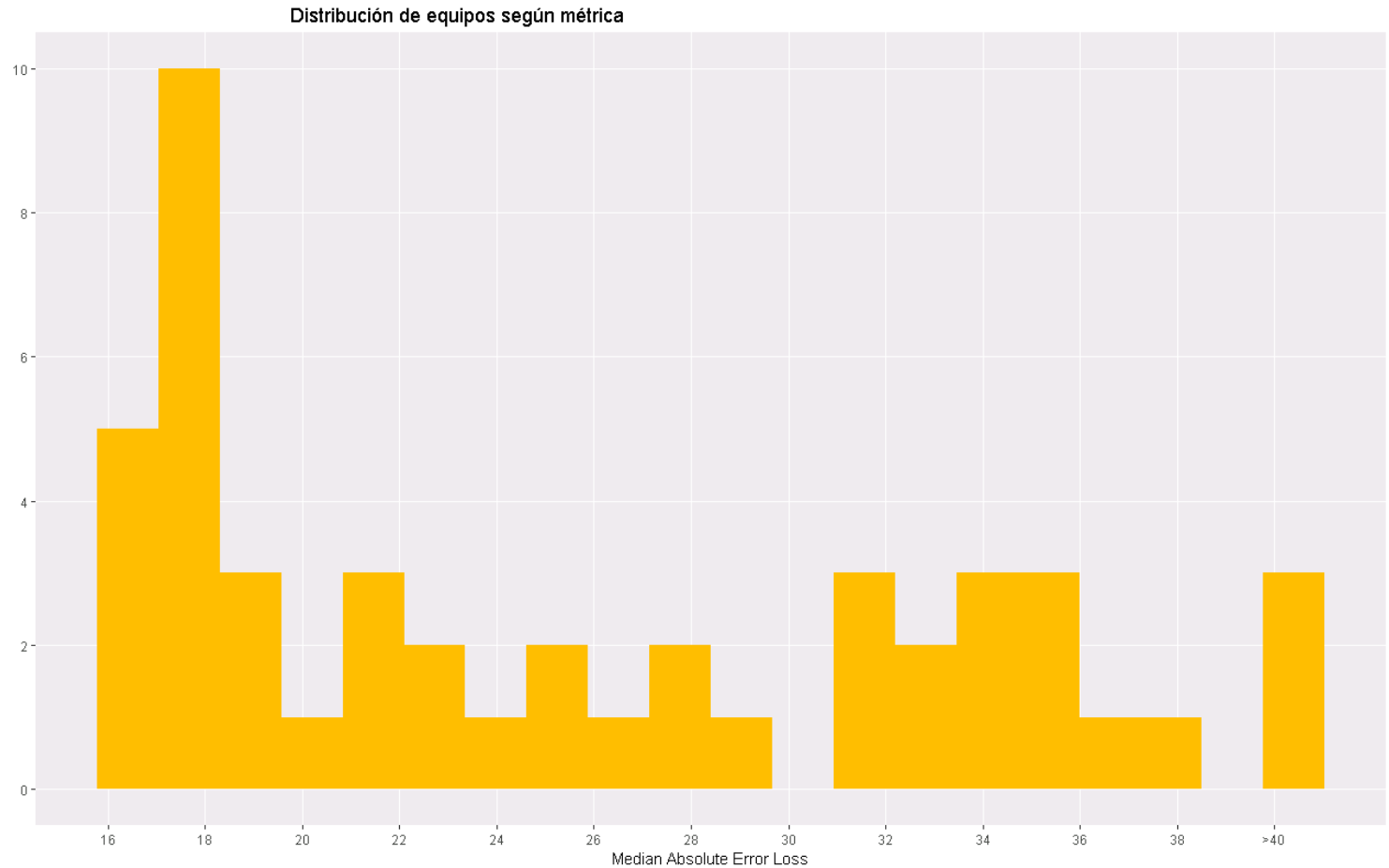

Figure 1: Distribución de resultados a nivel nacional en la fase local

Índice

Conclusiones

- Conocimiento sobre los datos que se va a trabajar
- Extracción y adecuación de las características más relevantes
- Pros/Cons de las distintas técnicas de ML
- **Ajuste de hiperparámetros**

Preguntas