TRABAJO DE FIN DE GRADO

Grado en Estadística

Título: Relevancia de variables en Redes Neuronales

Autor: Saray Sancho Morales

Director: Pedro Delicado Useros

Departamento: Estadística e Investigación Operativa

Convocatoria: Junio 2020

UNIVERSITAT DE BARCELONA

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat de Matemàtiques i Estadística

Resumen
Algunos algoritmos predictivos (como las redes neuronales) usualmente presentan mejores resultados en predicción que los modelos estadísticos que resuelven los mismos problemas (por ejemplo, el modelo de regresión lineal o GLM). Por el contrario, los modelos estadísticos son más fácilmente interpretables que los modelos algorítmicos porque ofrecen una medida de la contribución a la predicción que hace cada una de las variables explicativas. Este TFG parte del trabajo de Delicado and Peña (2019) titulado *Variable relevance by ghost variables* y quiere comparar estas medidas generales con las medidas que se han definido en la literatura para redes neuronales.

Palabras clave
Redes neuronales, *Neural Networks*,*Machine Learning*, *Deep Learning*, *Interpretable Machine Learning*

Clasificación AMS (American Mathematical Society)
62-04: Explicit Machine computation programs (not the theory of computation or programming)
62-07: Data analysis

Índice

1. Introducción

En los últimos años se ha experimentado un crecimiento del uso de algoritmos predictivos debido a sus buenos resultado a la hora de hacer estimaciones, especialmente en comparación con modelos estadísticos más clásicos. Debido a este crecimiento también se ha hecho notable la necesidad de comprender mejor cómo funcionan internamente y el por qué de las estimaciones que hacen ya que su naturaleza hace que sean modelos muy complicados de interpretar. Poder interpretar un modelo de *machine learning* es crucial para poder detectar sesgos en el rendimiento del modelo y conseguir que sus estimaciones generen una mayor confianza en sus resultados.

Este trabajo tiene como punto de partida el artículo *Variable relevance by ghost variables* (Delicado and Peña 2019) y busca poder comparar el método propuesto en dicho artículo con los métodos utilizados actualmente para determinar la relevancia que tiene una variable en una red neuronal.

Para este trabajo nos hemos focalizado en los métodos que miden la relevancia de una variable dentro del modelo aunque también existen métodos que se centran en explicar el por qué un determinado valor de una variable tiene una estimación determinada, esto se denomina análisis local, y no está incluido en el contenido de esta memoria.

Esta docuemnto está estructurado en dos bloques, el primero se centra en describir de una forma teórica cada uno de los métodos de interpretación, incluyendo el método de las variables fantasma. En el segundo bloque desarrollamos una red neuronal y probamos cada uno de los métodos detallados en el primer bloque con el objetivo de examinar su rendimiento y la diferente información que proporciona cada uno de ellos.

Finalmente, se detallan las conclusiones obtenidas tras todo el proceso de investigación y análisis.

2. Interpretación de Redes Neuronales - Métodos agnósticos de interpretación

A lo largo de este apartado se describen diferentes métodos para analizar la importancia que tienen las variables predictoras en el *output* de un modelo de *machine learning*. Tal y como indica el título de este apartado, los métodos explicados en él se pueden utilizar para interpretar la relevancia de las variables con independencia del modelo que se haya ajustado para realizar el ajuste de los datos. Para la realización de los ejemplos gráficos que acompañan a las explicaciones se ha utilizado en conjunto de datos *Boston Housing*, que forma parte del paquete *MASS* de R y contiene información sobre las viviendas de 506 distritos censales de Boston en el año 1970. En el *Anexo I* se puede consultar el código implementado para la obtención de los gráficos explicativos.

2.1. Relevancia por omisión

El cálculo de la relevancia mediante la omisión de variables consiste en ajustar un primer modelo con todas las variables del conjunto de datos que queremos estudiar y, a posteriori, otro excluyendo las variables cuya relevancia nos interesa.
La relevancia de una variable mediante omisión se calcula, tal que:

$$Rel_{OM}^{Train}(x_s) = \frac{1}{n_1}(\hat{y}_{1.X.x_s} - \hat{y}_{1.X})^T(\hat{y}_{1.X.x_s} - \hat{y}_{1.X}))$$

donde:

x_s representa las variables cuya relevancia se quiere obtener

$\hat{y}_{1.X.x_s}$ representa los valores estimados de la variable respuesta por el primer modelo ajustado

$\hat{y}_{1.X}$ representa los valores estimados de la variable respuesta por el segundo modelo

El conjunto x_s habitualmente esta formado por una o dos variables aunque este método permite incluir más.

Ventajas y desventajas
La principal desventaja de este método es el tiempo computacional que requiere entrenar dos redes neuronales distintas para poder compararlas. Además, la relevancia de una variable calculada mediante este método puede verse corrompida si existe una alta correlación entre la variable de interés y otra dentro del conjunto de datos; o bien si su interacción es significativa para el modelo. Este método puede parecer muy intuitivo a primera vista, pero el modelo con los datos reducidos no tiene sentido para la importancia de la variable. Estamos interesados en la importancia de las variables de un modelo fijo. Volver a entrenar con un conjunto de datos reducido crea un modelo diferente al que nos interesa.

Por otro lado, es un método muy sencillo de entender y su visualización también es especialmente fácil de interpretar.

2.2. Gráficos de dependencia parcial (PDP)

Los gráficos de dependencia parcial (PDP) muestran el efecto marginal que una o dos variables tienen sobre el resultado previsto de un modelo de aprendizaje automático (Friedman 2001). Además son capaces de representar qué tipo de relación existe entre ambas variables (ej. lineal). La función de dependencia parcial se calcula tal que:

$$\hat{f}_{x_s} = E_{x_c}\left[\hat{f}(x_s, x_c)\right] = \int \hat{f}(x_s, s_c) dP(x_s)$$

donde:

x_s representa las variables que se desean representar en PDP

x_c representa el resto de variables utilizadas en el modelo

$\hat{f}$ representa el modelo ajustado

Las variables incluidas en el conjunto x_s son aquellas cuyo efecto en la estimación queremos estudiar. Habitualmente esta formado por una o dos variables.
La función anterior es un promedio marginal de $\hat{f}$, y puede servir como una descripción útil de efecto del subconjunto elegido (x_s) cuando estas no tienen interacciones fuertes con las del subconjunto x_c. La funciones de dependencia parcial se pueden utilizar para interpretar los resultados de cualquier modelo *black-box*. Se pueden estimar así:

$$\bar{f}_s(x_s) = \frac{1}{N}\sum_{i=1}^{N} f(x_s, x_{ic})$$

donde:

$x_{ic} = \{x_{1c}, x_{2c}, ..., x_{nc}\}$ son los valores de x_c que están el los datos de entrenamiento

Este proceso requiere recorrer todo el conjunto de datos para cada conjunto de x_s evaluado en $\bar{f}_s(x_s)$. Este proceso puede ser lento y muy costoso computacionalmente, incluso para conjuntos de datos no muy extensos.
Es importante tener en cuenta que la función dependencia parcial definida anteriormente representa el efecto del conjunto x_s en $f(x)$ después de considerar el efecto promedio del conjunto de variables x_c en $f(x)$.

Ventajas y desventajas
Los gráficos de dependencia parcial son muy fáciles de interpretar ya que representa la estimación promedio si forzamos a todos los puntos de datos a asumir un valor en particular para la variable cuya importancia nos interesa.

Además, los gráficos de dependencia parcial son especialmente eficientes cuando la variable estudiada no está correlacionada con otras utilizadas en el modelo; permiten apreciar con claridad como el valor promedio de la estimación cambia cuando el valor de la variable de interés cambia.
Por otro lado, una de las desventajas de los gráficos de dependencia parcial es que únicamente se pueden estudiar dos variables, ya que resulta muy complejo plantear el gráfico con una dimensionalidad mayor.
Además, la principal desventaja de los gráficos de dependencia parcial es que asume independencia entre las variables. Este método asume que las variables que se estudian mediante un gráfico de dependencia parcial son independientes del resto de variables del modelo. De esta forma se ponderan elementos que en la realidad pueden no tener mucho sentido.
Finalmente, otra de las desventajas de este método es que las variables que tienen un efecto heterogéneo en la variable pueden interpretarse como no relevantes. Por ejemplo, si para una determinada variable los puntos en la mitad de su rango tienen una correlación positiva (cuanto mayor es al valor de la variable más aumenta el valor de la estimación), y los que están la otra mitad tiene una correlación negativa (cuanto menor es el valor que toma la variable más aumenta el valor de la estimación), es posible que el gráfico de dependencia parcial se muestre como una linea horizontal porque el efecto del conjunto de los datos puede cancelarse, dando lugar a interpretar que la variable apenas tiene efecto sobre la estimación que ha realizado el modelo.

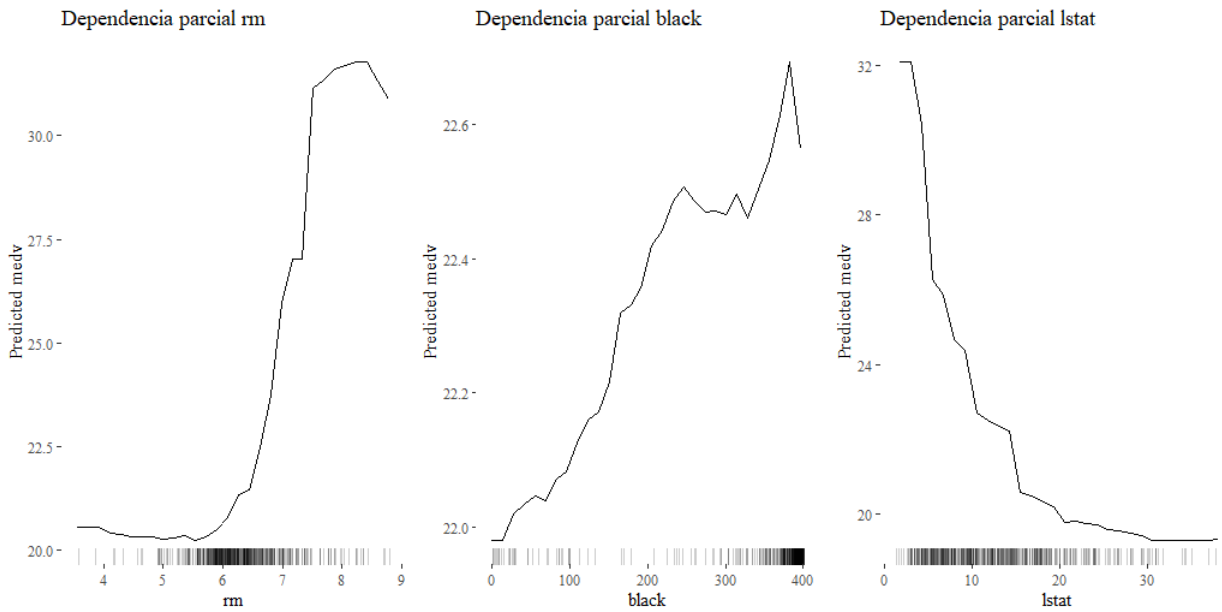


Figura 1: Ejemplos PDP Plot

2.3. Gráficos efectos locales acumulados (ALE)

Los efectos locales acumulados describen cómo las variables influyen de media en la estimación de un modelo de *machine learning*. Los gráficos ALE son una alternativa más rápida y menos sesgada a los gráficos de dependencia parcial (PDP).
Este método tiene el mismo objetivo que los anteriormente descritos gráficos de dependencia parcial: describir como una variable afecta a la estimación del modelo de media. Tal y como se ha especificado antes, si algunas variables utilizadas en el modelo están correlacionadas los gráficos de dependencia parcial son poco fiables ya que si se procesan con variables que tienen una fuerte correlación se calculará el promedio de la estimación incluyendo combinaciones de variables que son muy improbables en la realidad, constituyendo así un sesgo que puede afectar mucho a las predicciones.
A diferencia de los PDP los gráficos de efectos locales acumulados (ALE) promedian las predicciones condicionales para cada valor de la variable de interés que se encuentran dentro de un intervalo de valores predeterminado. Los gráficos ALE promedian los cambios en las predicciones y las acumulan a las de todo el margen:

$$\hat{f}_{x_s,ALE}(x_s) = \int_{z_{0,1}}^{x_s} E_{x_c|x_s}\left[\hat{f}(X_s, X_c)|X_s = z_s\right] dz_s - constante =$$

$$\int_{z_{0,1}}^{x_s} \int \hat{f}^s(z_s, x_c) P(x_c|z_s) dx_c dz_s - constante$$

Tal y como se puede observar en las fórmulas de ambos gráficos su principal diferencia está en la distribución que utilizan para su cálculo. Los gráficos de dependencia parcial (PDP) utilizan la distribución marginal para el cálculo de la relevancia mientras que los gráficos de efectos acumulados (ALE) utilizan la distribución condicional.
Por este motivo, cuando las variables predictoras están altamente correlacionadas la distribución marginal utilizada por los gráficos de dependencia parcial (PDP) es más amplia que la distribución condicional y puede incluir áreas de estimación donde realmente no existen datos.
La distribución condicional utilizada por los gráficos ALE ayuda a mitigar este problema, lo que puede hacer que este método sea preferible en casos en que las predictoras están altamente correlacionadas.

Ventajas y desventajas
La principal ventaja de los gráficos de efectos locales acumulados es que son menos sensibles a la correlación entre las variable predictoras que otros métodos.
Otras de sus ventajas son una rápida velocidad de computación, al limitar el número de puntos estimados mediante intervalos; y la fácil interpretación del gráfico resultante.

Sin embargo, estos gráficos pueden ser muy complejos de implementar. Además tampoco existe un método que permita decidir el número de intervalos óptimo para realizar el gráfico. Y, cuando se ejecuta para dos variables predictoras puede resultar confuso, ya que el *heatmap* que se obtiene puede dar la sensación de representar el efecto total de ambas variables sobre el *output* mientras que lo que representa es el efecto adicional de la interacción.

Finalmente, a pesar de ser un método más fiable cuando existe correlación entre las variables predictoras su interpretación sigue siendo complicada cuando dos de ellas tienen una alta correlación.

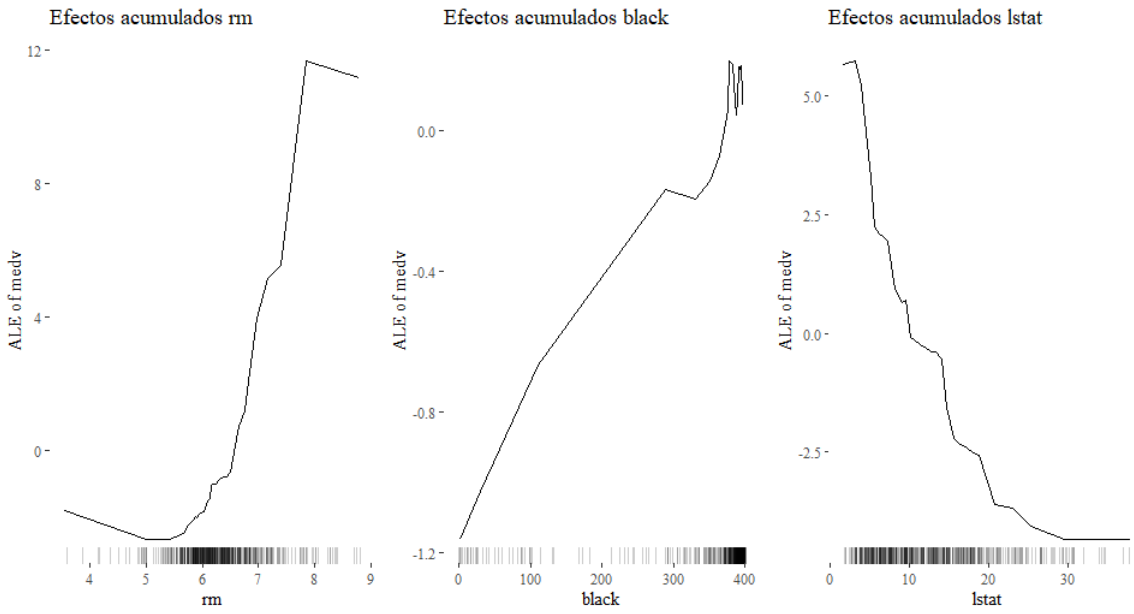


Figura 2: Ejemplos ALE Plot

2.4. Interaccíon de variables

Cuando existe cierta correlación entre las variables de un modelo predictivo, la estimación no puede expresarse como la suma de los efectos de estas, porque el efecto de una de esas variables depende del valor de otra de ellas.
Si un modelo de *machine learning* realiza una estimación basada en dos variables, podemos descomponer la estimación en cuatro términos: un término constante, un término para la primera variable, un término para la segunda variable y un término para la interacción entre ambas. La interacción entre dos variables es el cambio que se da en la estimación después de considerar los efectos individuales de las variables.
Una forma de estimar la intensidad de la interacción es medir qué parte la variación de la estimación depende de la interacción de las variables. Este proceso se puede realizar mediante el estadístico H de Friedman (Friedman 2001). Se pueden dar dos casos:

- Una medida de interacción bidireccional que nos dice si dos variables del modelo interactúan entre sí y en qué medida lo hacen.
- Una medida de interacción total que nos dice si, y en qué medida, una variable interactúa en el modelo con todas las demás.

Si no existe interacción entre las dos variables, podemos descomponer la función de dependencia parcial de la siguiente manera (asumiendo que las funciones de dependencia parcial estén centradas en cero):

$$PD_{jk}(x_j, x_k) = PD_j(x_j) + PD_k(x_k)$$

donde:

$PD_{jk}(x_j, x_k)$ es la función de dependencia parcial para ambas variables (j,k)

$PD_j(x_j)$ es la función de dependencia parcial de la variable j, únicamente

$PD_k(x_k)$ es la función de dependencia parcial de la variable k, únicamente

Igualmente, si una determinada variable no tiene interacción con ninguna otra la función $\hat{f}(x)$ se puede expresar como una suma de funciones de dependencia parcial, donde el primer sumando depende exclusivamente de la variable que estamos estudiando (j) y el segundo de todas las demás:

$$\hat{f}(x) = PD_j(x_j) + PD_{-j}(x_{-j})$$

donde:

$PD_{-j}(x_{-j})$ es la función de dependencia parcial que depende de todas las variables a excepción de j

Esta descomposición expresa la función de dependencia parcial (o estimación completa) sin interacciones (entre las variables j y k, o respectivamente j y todas las demás).
El siguiente paso consiste en medir la diferencia entre la función de dependencia parcial observada y las descompuesta sin interacciones. Se calcula la varianza del resultado de la función de dependencia parcial (para medir la interacción entre dos variables) o de la función completa (para medir la interacción entre una característica y todas las demás variables).

La cantidad de varianza explicada por la interacción (diferencia entre la dependencia parcial observada y sin interacción) se utiliza para medir la fuerza de la interacción. Esta diferencia toma, habitualmente, valores entre cero y uno, donde cero representa la ausencia de interacción y uno manifiesta que cada función de dependencia es constante y viene dada únicamente por interacción. Ocasionalmente este estadístico toma valores superiores a uno, cuando esto ocurre su interpretación resulta mucho más compleja.

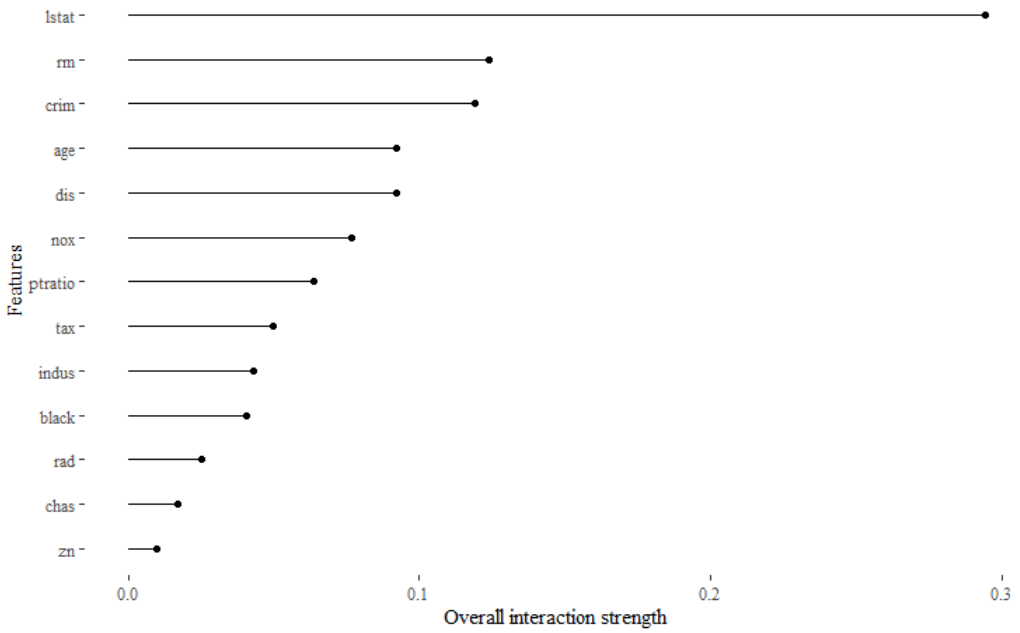


Figura 3: Ejemplo gráfico de interacción global

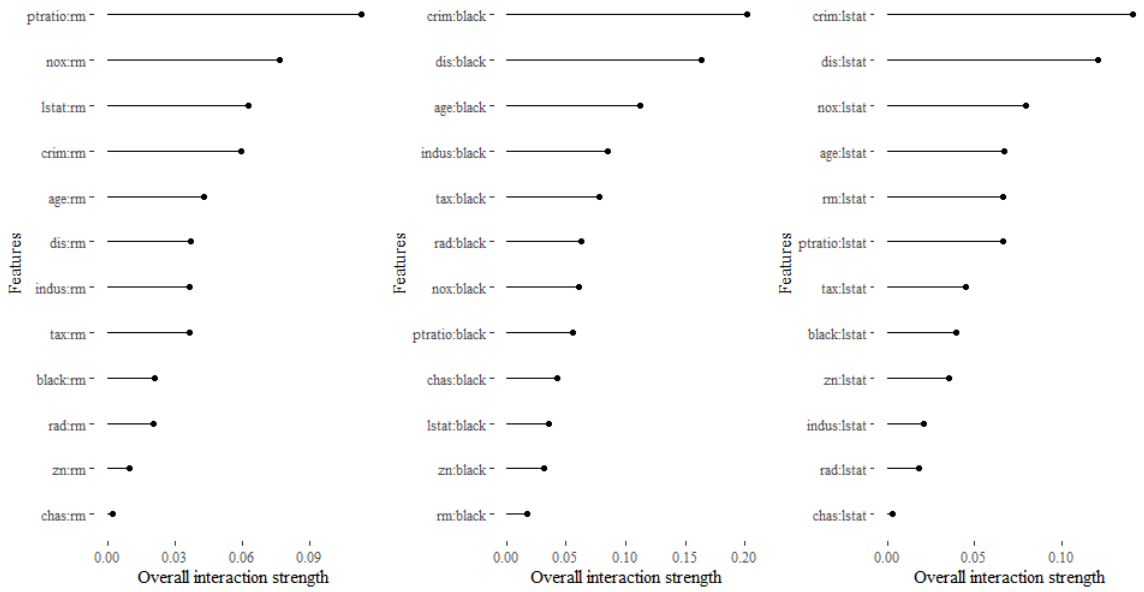


Figura 4: Ejemplo gráfico de interacción de las variables *rm*, *black* y *lstat*

2.5. Método de las permutaciones

El método de las permutaciones mide el aumento en el error de estimación del modelo después de permutar los valores de la variable de interés, lo que rompe la relación entre esa variable y el output.

Una variable es relevante si al permutar sus valores aumenta el error del modelo, porque en este caso el output obtenido fue en gran parte debido a la variable de interés. Una variable no es

importante si al permutar sus valores deja el error del modelo sin cambios, porque en este caso el modelo ignoró la variable para la estimación. Fisher, Rudin, and Dominici (2018) propusieron este método independiente del modelo para medir la importancia de las variables; también introdujeron ideas más avanzadas sobre la importancia de las variables como, por ejemplo, una versión (específica del modelo) que tiene en cuenta que muchos modelos de estimación pueden predecir bien los datos.

Dados:

f modelo de red neuronal que se desea estudiar

X matriz del conjunto de variables predictoras del modelo

y vector de valores de la variable respuesta

L(y,f) medida del error de estmación del modelo

El algorítmo de permutación de variables es tal que:

1. Estimar el error del modelo original $e^{original} = L(y, f(x))$
2. Para cada una de las variables $j = 1, ..., p$ de la la matrix X:
   - Generar la nueva matrix $X^{permutada}$, permutando los valores la variable j en la matriz X.
   - Estimar el error $e^{permutado} = L(y, f(X^{permutada}))$.
   - Calcular la importancia de la variable:

$$FI_j = \frac{e^{permutado}}{e^{original}}$$

3. Ordenar el conjunto de variables por FI descendente

Finalmente, obtenemos un vector de importancias para cada una de la variables j analizadas, que se puede representar mediante un gráfico para facilitar su interpretación.

Ventajas y desventajas

La principal ventaja de este método es su fácil interpretación, se mide en función del aumento en el error del output cuando la información que aporta esta variable es completamente eliminada.

Un aspecto positivo del uso de la diferencia porcentual del error en lugar de la diferencia absoluta es que la medida de relevancia de diferentes variables se puede comparar entre sí y también con diferentes problemas. Esta medida tiene en cuenta automáticamente todas las interacciones con otras variables. Al permutar los valores, también destruye los efectos de interacción con otras variables. Esto significa que la importancia de la variable que se permuta tiene en cuenta tanto el efecto de esta variable como los efectos de interacción en la precisión del modelo. Esto

también es una desventaja porque la importancia de la interacción entre dos variables está incluida en las medidas de importancia de ambas características.
A diferencia del cálculo de relevancia mediante omisión la importancia de la variable mediante permutación no requiere volver a entrenar el modelo. Dado que el reciclaje de un modelo de aprendizaje automático puede llevar mucho tiempo, permutar solo una variable puede ahorrar mucho tiempo.

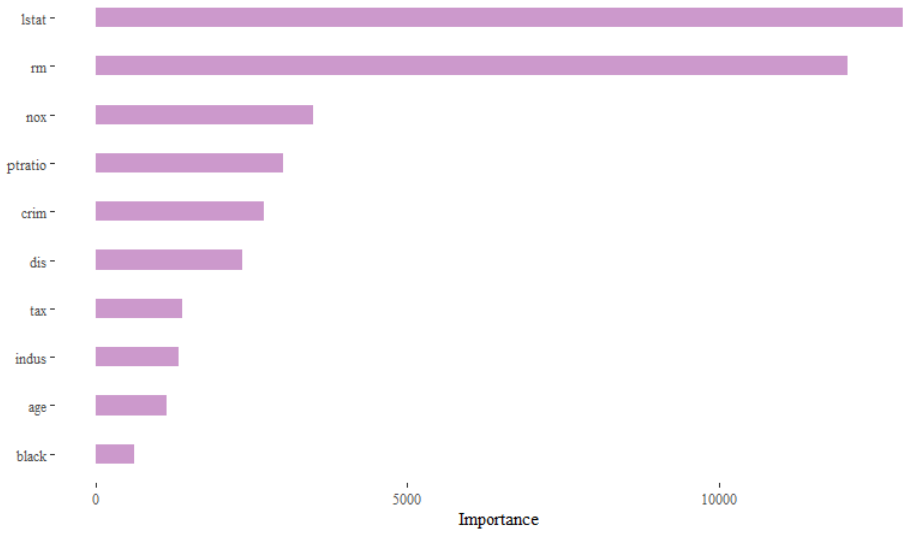


Figura 5: Ejemplo gráfico de relevancia mediante el método de las permutaciones

3. Método de las variables fantasma

Tal y como hemos ido desarrollando a lo largo de este trabajo, los métodos para analizar la relevancia de una variable en un red neuronal son bastante complejos y tienen como principales desventajas el tiempo o coste de reajustar más de un modelo o la pérdida de información cuando algunas de las variables respuesta tienen una alta correlación entre sí.

En este apartado vamos a describir el método propuesto en el artículo (Delicado and Peña 2019).

En el citado artículo se propone estimar $E[Z|X]$, donde Z representa la variable cuya relevancia queremos estudiar y X el resto de variables explicativas del conjunto de datos, ajustando un modelo predictivo muy sencillo (un modelo de regresión lineal, por ejemplo) utilizando los datos de test del modelo. Para explicar el método vamos a suponer que la función de regresión es lineal y se ha estimado mediante mínimos cuadrados ordinarios.
Si tenemos β_x y β_z como los coeficientes estimados del modelo y $\hat{\sigma}^2$ la varianza residual estimada, ajustamos un modelo de regresión mediante mínimos cuadrados ordinarios el conjunto de *test* para obtener los valores fantasma para nuestra variable de intrés x_2:

$$\hat{z}_{2,2} = X_2\hat{\alpha}_2$$

donde $\hat{\alpha_2} = (X_2^T X_2)^{-1} X_2^T z_2$ y, por tanto:

$$\hat{y}_{2.X.z} = X_2\hat{\beta}_x + z_2\hat{\beta}_z$$

cuando se usan X y Z como variables predictoras, y:

$$\hat{y}_{2.X.\hat{z}} = X_2\hat{\beta}_x + \hat{z}_{2,2\hat{\beta}_z}$$

usando X y $\hat{Z}_X$, remplazando a la variable Z por sus valores fantasma. De este modo la relevancia dada mediante el método de las variables fantasma se mide como:

$$Rel_{fantasma}(Z) = \frac{1}{n_2}(\hat{y}_{2.X.z} - \hat{y}_{2.X.\hat{z}})^T(\hat{y}_{2.X.z} - \hat{y}_{2.X.\hat{z}})$$

4. Comparación de los métodos de interpretación habituales con el método de las variables fantasma

4.1. Conjunto de datos de ejemplo

Este apartado del trabajo tiene como objetivo utilizar y comparar los diferentes métodos explicados anteriormente de una forma práctica. Para ello utilizaremos un conjunto de datos reales sobre viviendas de alquiler procedentes de la web Idealista ()`www.idealista.com`) que permite a los usuarios buscar viviendas de alquiler filtrando por diferentes campos (precio máximo, m2, localización, número de habitaciones,...) entre las ofertas publicadas por otros usuarios.

Estos datos se obtuvieron desde la página web de Idealista por Alejandro Germán-Serrano el 27 de Febrero de 2018 y fueron publicados en la dirección web `https://github.com/seralexger/idealista-data;`. Actualmente no se encuentran disponibles para su descarga y han sido facilitados para este trabajo por el profesor Pedro Delicado.

El conjunto de datos está formado por un total de 16480 registros de Madrid y Barcelona y 17 variables:

price precio mensual del alquiler, variable objetivo

Barcelona indicador de si la población es Barcelona. Toma valores 0-1.

categ.distr indicador del precio del área de la vivienda. Toma valores 1, 2, 3 y 4.

type.chalet indica si la vivienda es una casa. Toma valores 0-1.

type.duplex indica si la vivienda es un duplex. Toma valores 0-1.

type.penthouse indica si la vivienda es un ático. Toma valores 0-1.

type.studio indica si la vivienda es un estudio. Toma valores 0-1.

floor piso de la vivienda.

hasLift indica si la vivienda tiene ascensor. Toma valores 0-1.

floorLift abs(floor)*(1-hasLift).

size tamaño de la vivienda en metros cuadrados.

exterior indica si la vivienda es exterior. Toma valores 0-1.

rooms número de dormitorios de la vivienda.

bathrooms número de baños de la vivienda.

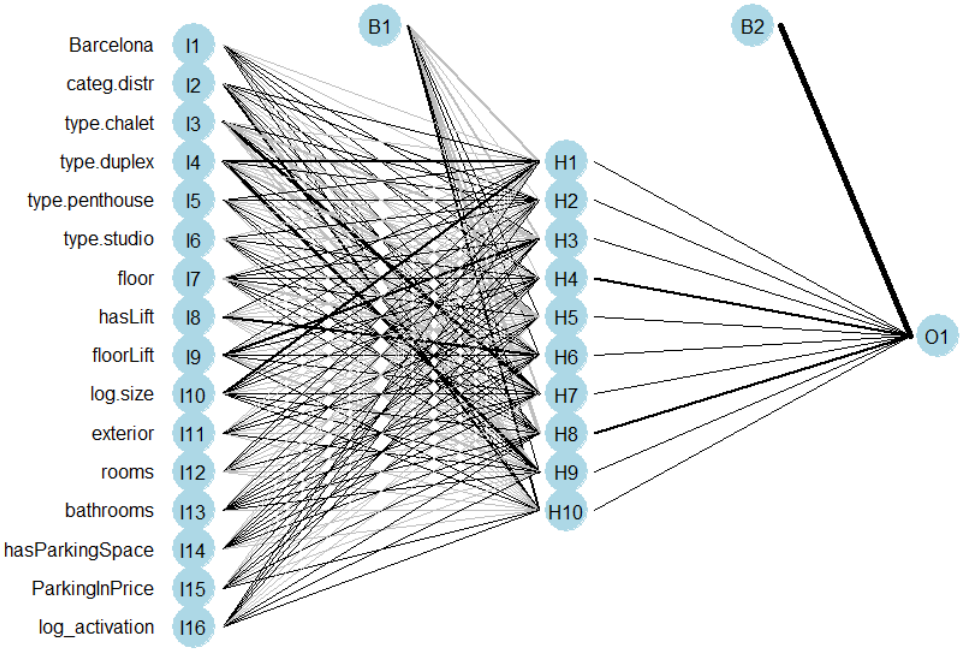
hasParkingSpace indica si la vivienda tiene aparcamiento. Toma valores 0-1.

ParkingInPrice indica si el aparcamiento se incluye en el precio mensual de la vivienda. Toma valores 0-1.

log_activation logaritmo del número de días que pasaron desde la primera activación del anuncio en la web.

En el *Anexo II* se puede consultar en análisis descriptivo de este conjunto de datos. Para realizar todos los análisis hemos ajustado una red neuronal utilizando los paquetes *caret* y *nnet* de R. El *summary* del modelo tiene el siguiente *output*:

```
a 16-10-1 network with 181 weights inputs: Barcelona categ.distr
type.chalet type.duplex type.penthouse type.studio floor hasLift
floorLift log.size exterior rooms bathrooms hasParkingSpace
ParkingInPrice log_activation output(s): .outcome options were -
linear output units
```



4.2. Métodos de interpretación agnósticos

4.2.1. Relevancia por omisión

Para ejemplificar este método calculamos la relevancia por omisión de las variables *log.size*, *categ.distr*, *type.chalet* y *hasLift*. También calcularemos la relevancia por omisión de dos variables a la vez, *hasLift* y *type.chalet*.

Los resultados obtenidos han sido:

	Relevancia
log.size	0.0302
type.chalet	0.0037
categ.distr	0.0111
hasLift	0.0042
hasLift & type.chalet	0.0064

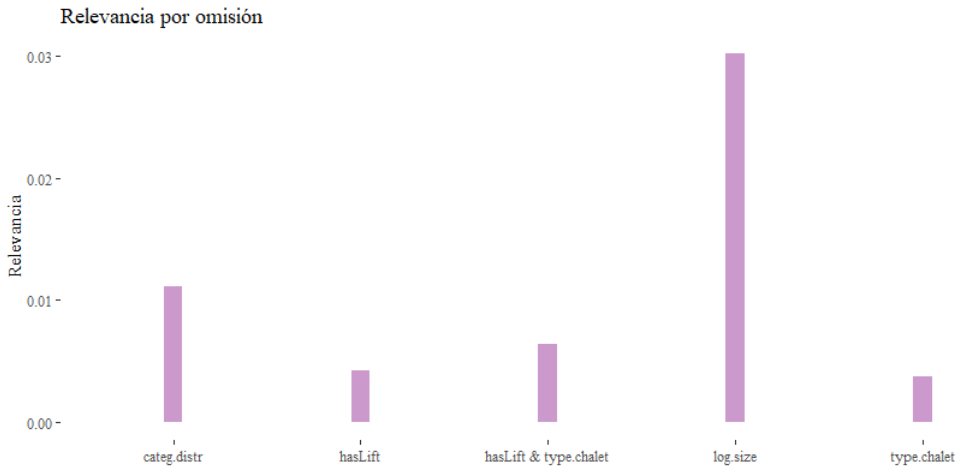


Figura 6: Relevancia por omisión de las variables *log.size*, *type.chalet*, *categ.distr*, *hasLift* y *hasLift & type.chalet*

Calculando la relevancia mediante omisión obtenemos que la variable más relevante es *log.size*, seguida de *categ.distr*. También destaca que si omitimos las variables *hasLift* y *type.chalet* a la vez su relevancia no aumenta de forma proporcional por lo que podemos asumir que ni la información que aportan estas variables ni su interacción aportaban mucho valor a la red neuronal.

4.2.2. Gráficos de dependencia parcial (PDP)

Para la realización de los gráficos de esta sección se ha utilizado la función `FeatureEffect$new` del paquete *iml* implementado en R. Es necesario tener en cuenta para analizar estos gráficos que la variable *target* de nuestro modelo es el logaritmo de la variable *price*, por lo que para

poder medir el efecto que tienen sobre el precio del alquiler tendremos que reinterpretar las variaciones que se producen en los gráficos.
A continuación se muestran los gráficos de dependencia parcial para cada una de las variables del conjunto de datos analizado:

Barcelona

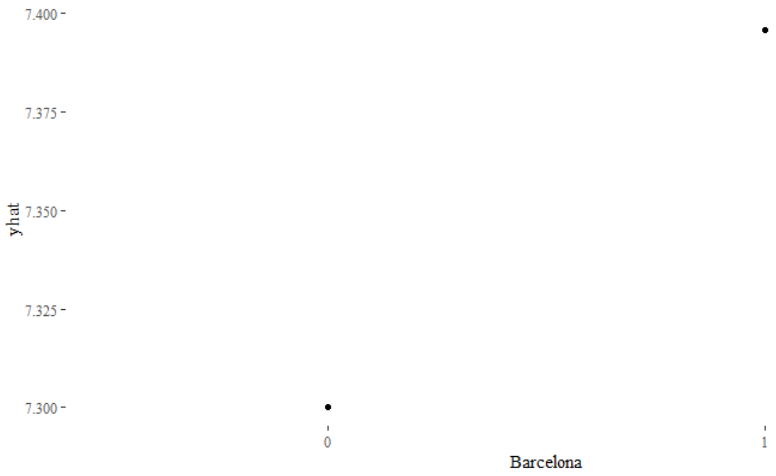


Figura 7: PDP Plot *Barcelona*

En este caso la variación que se produce sobre el precio de la vivienda es:

- Cuando la variable toma valor 0 la estimación del precio del alquiler es $\rightarrow e^{7,403} = 1479{,}66€$
- Cuando la variable toma valor 1 la estimación del precio del alquiler es $\rightarrow e^{7,403} = 1642{,}50€$

Esto supone una variación de 162.84€en la estimación del precio del alquiler de la vivienda.

Categ.distr

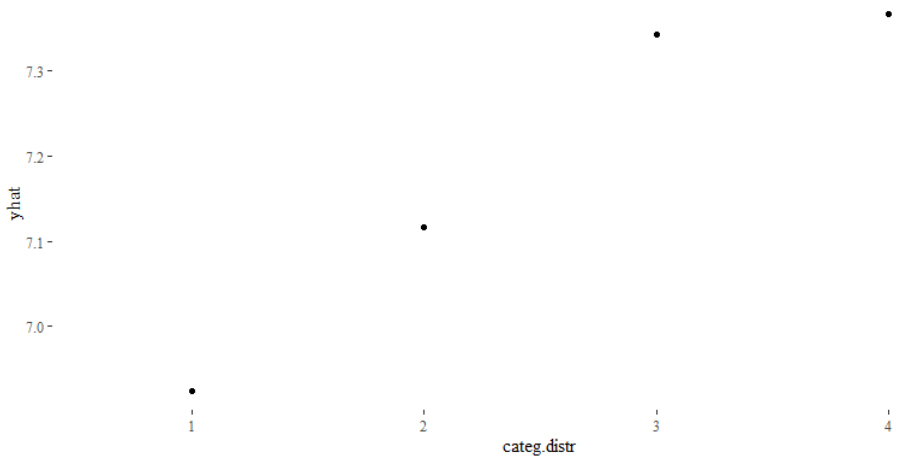


Figura 8: PDP Plot *categ.distr*

- Cuando la variable toma valor 1 la estimación del precio del alquiler es de 1053.15€
- Cuando la variable toma valor 2 la estimación del precio del alquiler es de 1287.69€
- Cuando la variable toma valor 3 la estimación del precio del alquiler es 1584.61€
- Cuando la variable toma valor 4 la estimación del precio del alquiler es 1595.38€

La diferencia entre la primera categoría y la última es de 542.23€, lo que supone una variación bastante grande.
Sin embargo, la diferencia entre las dos últimas categorías (valores 3 y 4) es bastante inferior, este cambio supone un incremento menor a 15€en el precio del alquiler de la vivienda.

Type.chalet, type.duplex, type.penthouse y type.studio

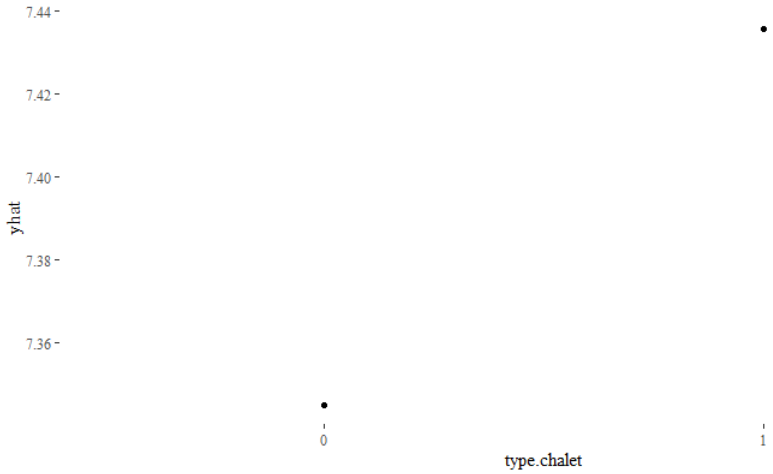


Figura 9: PDP Plot *type.chalet*

- Cuando la variable toma valor 0 la estimación del precio del alquiler es → $e^{7,346}$ = 1551,44€
- Cuando la variable toma valor 1 la estimación del precio del alquiler es → $e^{7,454}$ = 1727,94€

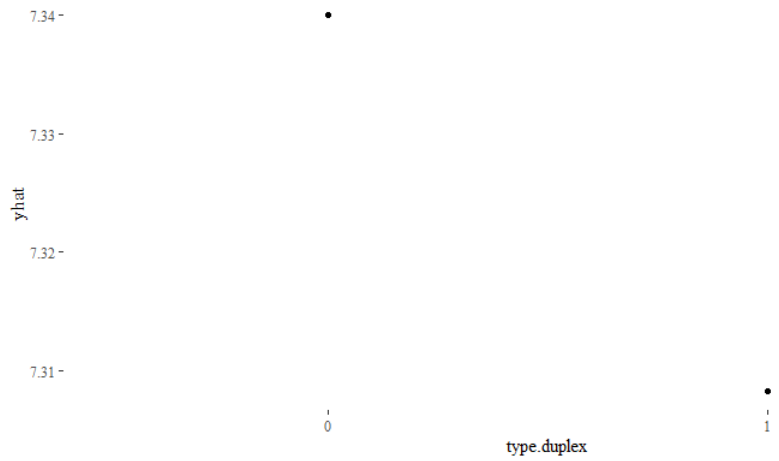


Figura 10: PDP Plot *type.duplex*

- Cuando la variable toma valor 0 la estimación del precio del alquiler es → $e^{7,341}$ = 1542,44€
- Cuando la variable toma valor 1 la estimación del precio del alquiler es → $e^{7,944}$ = 1623,99€

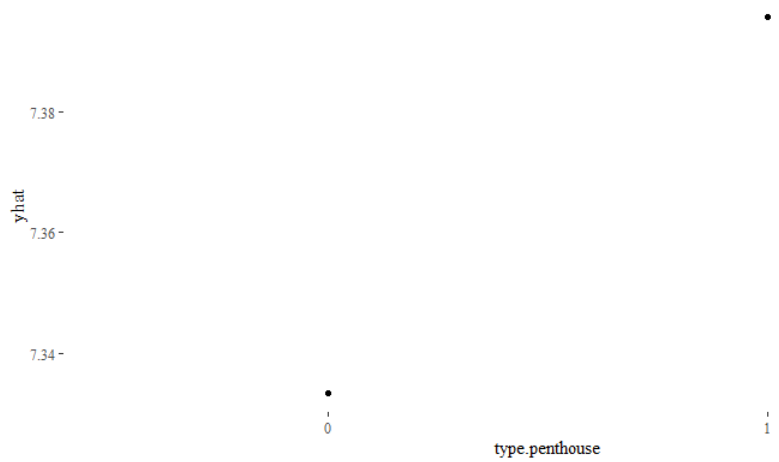


Figura 11: PDP Plot *type.penthouse*

- Cuando la variable toma valor 0 la estimación del precio del alquiler es → $e^{7,336}$ = 1534,92€

- Cuando la variable toma valor 1 la estimación del precio del alquiler es → $e^{7,424}$ = $1676,87€$

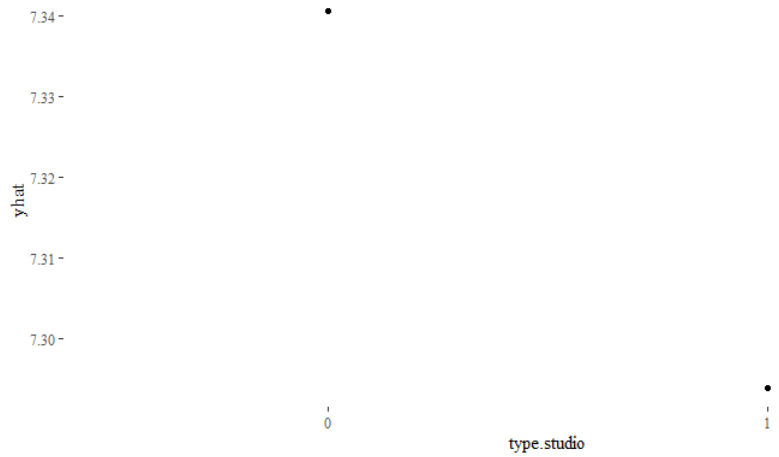


Figura 12: PDP Plot *type.studio*

- Cuando la variable toma valor 0 la estimación del precio del alquiler es → $e^{7,343}$ = $1546,25€$
- Cuando la variable toma valor 1 la estimación del precio del alquiler es → $e^{7,286}$ = $1459,79€$

Una de las cosas más relevantes de estas cuatro variables es que todas ellas, a excepción de *type.studio* suponen un incremento en el precio del alquiler. La variable que cuando toma valor uno supone una mayor variaicón en el precio del alquiler, de 176.5€, es *type.chalet* y la que implica una menor variación en el precio es la variable *type.duplex*, que solo incrementa el precio en 81.55€.
Por tanto, de estas cuatro variables la que tiene una mayor relevancia es *type.chalet*.

Floor

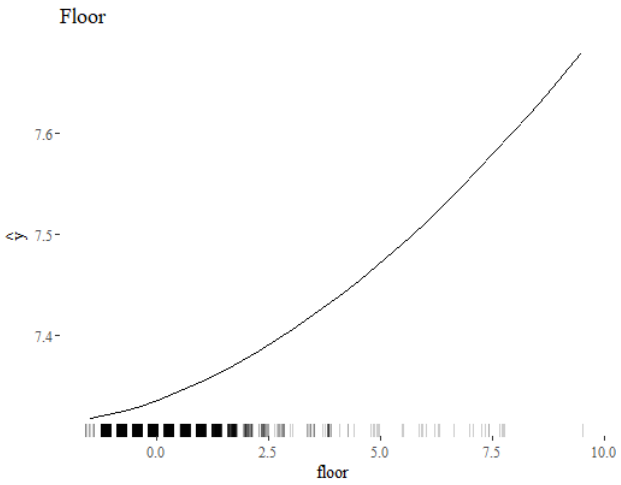


Figura 13: PDP Plot *floor*

- Cuando la variable toma su valor más pequeño la estimación del precio del alquiler es → $e^{7,318}$ = 1507,19€
- Cuando la variable toma su valor más alto la estimación del precio del alquiler es → $e^{7,677}$ = 2159,48€

En este caso, la variación que se produce en la estimación del precio del alquiler es bastante grande.

HasLift

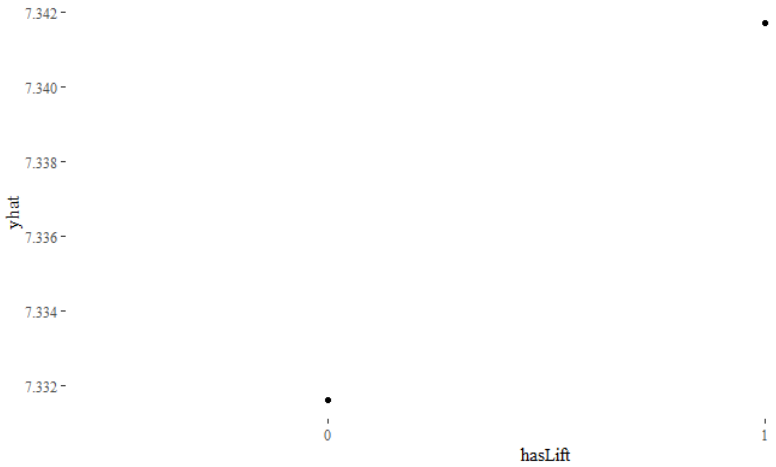


Figura 14: PDP Plot *hasLift*

- Cuando la variable toma su valor más pequeño la estimación del precio del alquiler es → $e^{7,318}$ = 1507,19€

- Cuando la variable toma su valor más alto la estimación del precio del alquiler es → $e^{7,677}$ = 2159,48€

La variación en la estimación del precio del alquiler cuando esta variable toma valor 1 es de 136.13€.

FloorLift

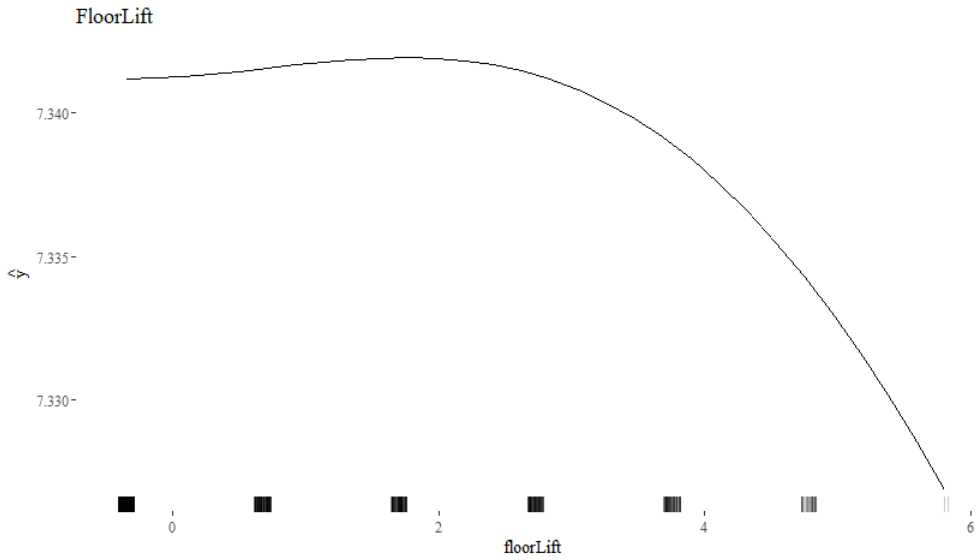


Figura 15: PDP Plot *floorLift*

- Cuando la variable toma su valor más pequeño la estimación del precio del alquiler es → $e^{7,327}$ = 1542,25€
- Cuando la variable toma su valor más alto la estimación del precio del alquiler es → $e^{7,341}$ = 1519,292€

En esta caso la variación en la estimación del precio del alquiler se mantiene más o menos constante a excepción de cuando la variable *floorLift* toma valores mayores a 4, que es cuando la estimación del modelo empieza a descender.

Exterior

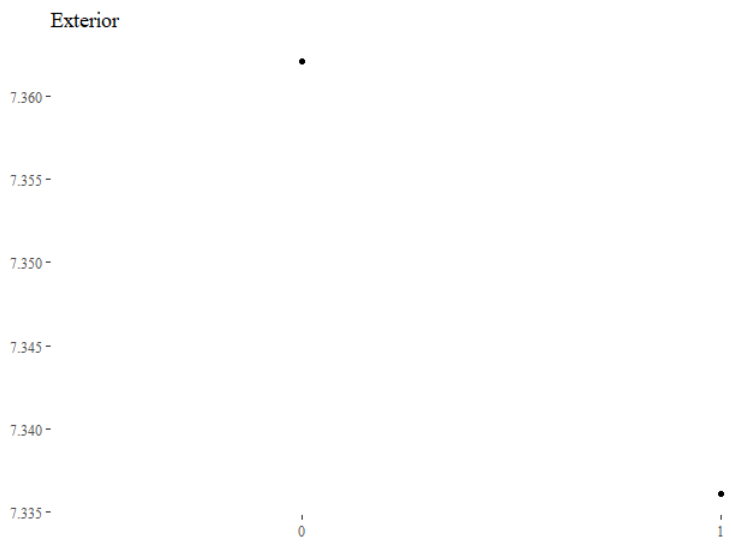


Figura 16: PDP Plot *exterior*

- Cuando la variable toma valor 0 la estimación del precio del alquiler es → $e^{7,362}$ = $1574,98€$
- Cuando la variable toma valor 1 la estimación del precio del alquiler es → $e^{7,336}$ = $1534,69€$

El resultado de esta variable es particularmente contra-intuitivo ya que explica que la estimación que hace la red neuronal de un piso que sí es exterior es inferior respecto a uno que no lo sea cuando lo natural sería pensar lo contrario.

Rooms

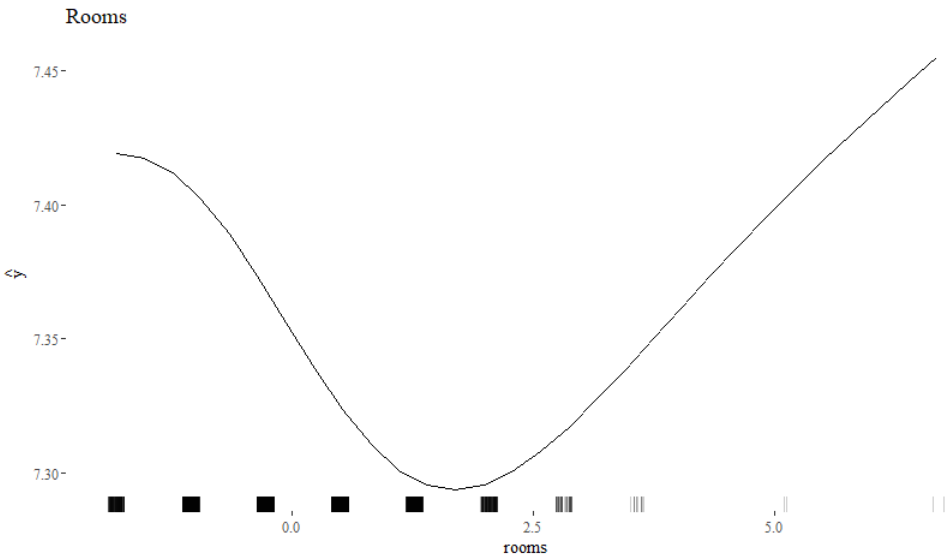


Figura 17: PDP Plot *rooms*

- Cuando la variable toma su valor más pequeño la estimación del precio del alquiler es → $e^{7,327}$ = $1667,49€$
- Cuando la variable toma su valor medio la estimación del precio del alquiler es es → $e^{7,327}$ = $1469,97€$
- Cuando la variable toma su valor más alto la estimación del precio del alquiler es → $e^{7,341}$ = $1728,24€$

Según el resultado obtenido podemos decir que según este método cuando la variable *rooms* toma valores de 0 a 4 el valor de la estimación del precio del alquiler decrece con cada habitación mientras que cuando la variable toma valor 5 o mayor la estimación del precio se vuelve a incrementar.

Bathrooms

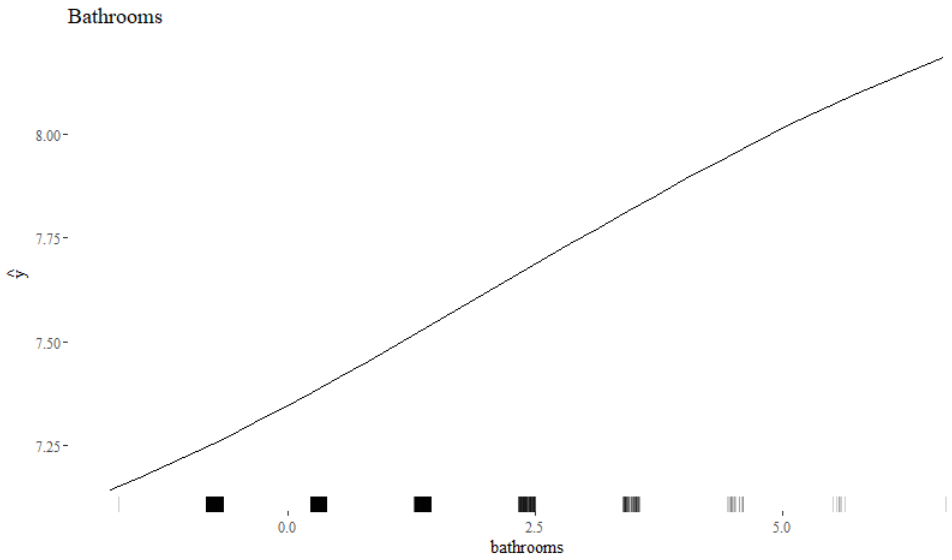


Figura 18: PDP Plot *bathrooms*

- Cuando la variable toma su valor más pequeño la estimación del precio del alquiler es → $e^{7,143}$ = 1265,46€
- Cuando la variable toma su valor más alto la estimación del precio del alquiler es → $e^{8,184}$ = 3583,15€

En este caso, la variable *bathrooms* produce una variación bastante grande, superior a los 1000€desde su valor más bajo hasta su valor más alto, en la estimación que hace la red neuronal sobre el precio del alquiler de la vivienda. Podemos concluir que según el método de ls gráficos de dependencia parcial la variable *bathrooms* es relevante para el modelo

Log.size

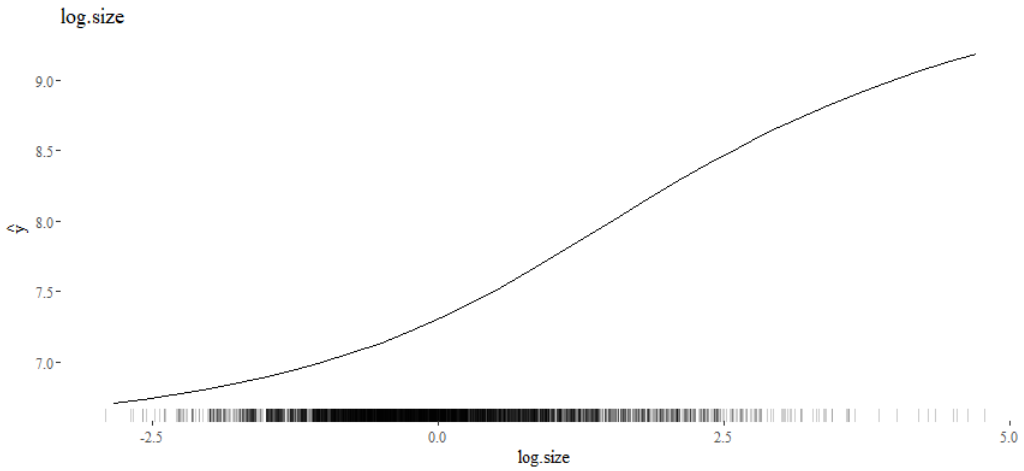


Figura 19: PDP Plot *log.size*

Esta variable es, sin duda, la que genera un mayor cambio en el *output* del modelo. Utilizando el método de las dependencias parciales es la variable más relevante de todo el conjunto de datos.

- Cuando la variable toma su valor más pequeño la estimación del precio del alquiler es → $e^{6,660}$ = 781,21€
- Cuando la variable toma su valor más alto la estimación del precio del alquiler es → $e^{9,304}$ = 10982,85€

La variación en función del valor de esta variable en la estimación del precio del alquiler es muy grande.

HasParkingSpace

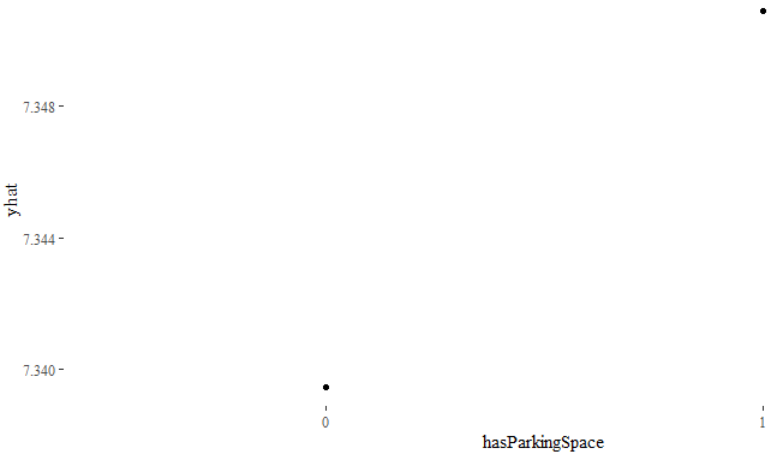


Figura 20: PDP Plot *hasParkingSpace*

- Cuando la variable toma valor 0 la estimación del precio del alquiler es → $e^{7,338}$ = $1537,63€$
- Cuando la variable toma valor 1 la estimación del precio del alquiler es → $e^{7,351}$ = $1557,75€$

En este caso la vaición que genera el cambio de valor de la variable *hasParkingSpace* es bastante pequeña, especialmente en comparación con el resto de variables analizadas.

ParkingInPrice

Figura 21: PDP Plot *ParkingInPrice*

- Cuando la variable toma valor 0 la estimación del precio del alquiler es → $e^{7,358}$ = $1568,70€$
- Cuando la variable toma valor 1 la estimación del precio del alquiler es → $e^{7,308}$ = $1492,19€$

En este caso cuando la variable *ParkingInPrice* el valor de la estimación del precio del alquiler disminuye en 76.51€. Este resultado es particularmente interesante porque según el gráfico de dependencia parcial cuando esta variable toma valor 1 el precio de la vivienda es ligeramente inferior cuando lo más intuitivo sería que fuese al revés.

Log_activation

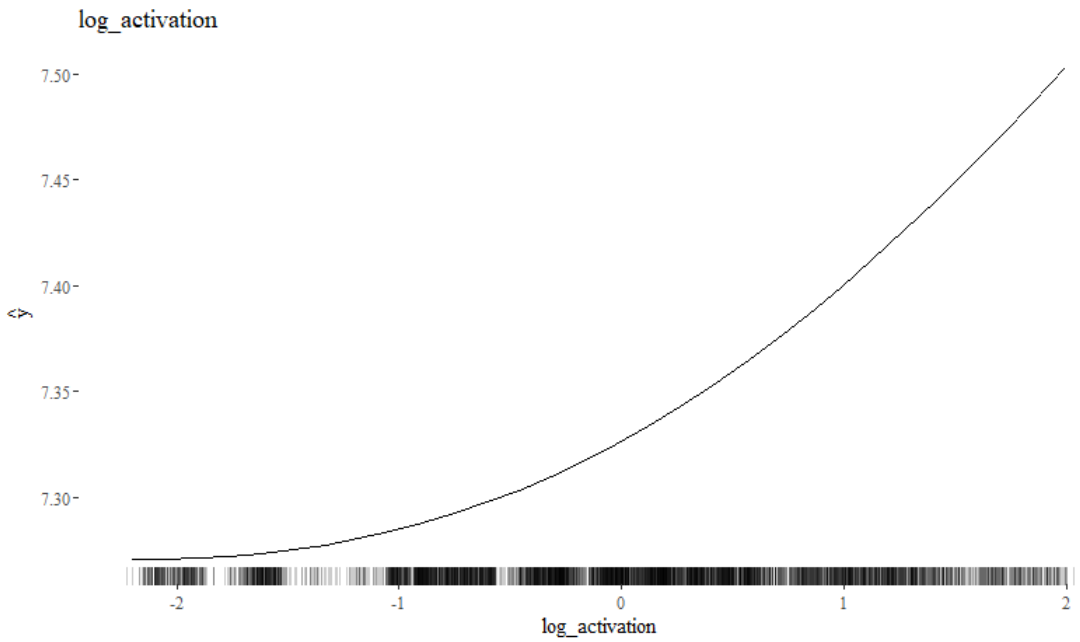


Figura 22: PDP Plot *log_activation*

- Cuando la variable toma su valor más pequeño la estimación del precio del alquiler es → $e^{7,270}$ = 1437,38€
- Cuando la variable toma su valor más alto la estimación del precio del alquiler es → $e^{7,503}$ = 1814,08€

En este caso la variación en el valor de la estimación que realiza el modelo también es bastante grande y esta aumenta cuando se incrementa el valor de la variable *log_activation*.

Resumen de resultados

Según el método de los gráficos de dependencia parcial podemos concluir que la variable más relevante es *log.size*, seguida de *bathrooms*, *floor* y *floorLift*, ya que son las que producen una mayor variación en la estimación que realiza la red neuronal del precio del alquiler de la vivienda. También es relevante que las variables *type.studio*, *exterior* y *ParkingInPrice* tienen un efecto disminuyen el valor de la estimación del modelo, las viviendas que son estudios , son exteriores o cuyo parking está incluido en el alquiler son más baratas. Por otro lado, las variables con una menor relevancia según este método son *floorLift*, *exterior* y *hasParkingSpace* que producen las variaciones más pequeñas en la estimación del precio del alquiler que realiza la red neuronal.

4.2.3. Gráficos de efectos acumulados (ALE)

Para la realización de los gráficos de esta sección se ha utilizado la función `FeatureEffect$new` del paquete *iml* implementado en R.
A continuación se muestran los gráficos de efectos acumulados para cada una de las variables del conjunto de datos analizado:

Barcelona

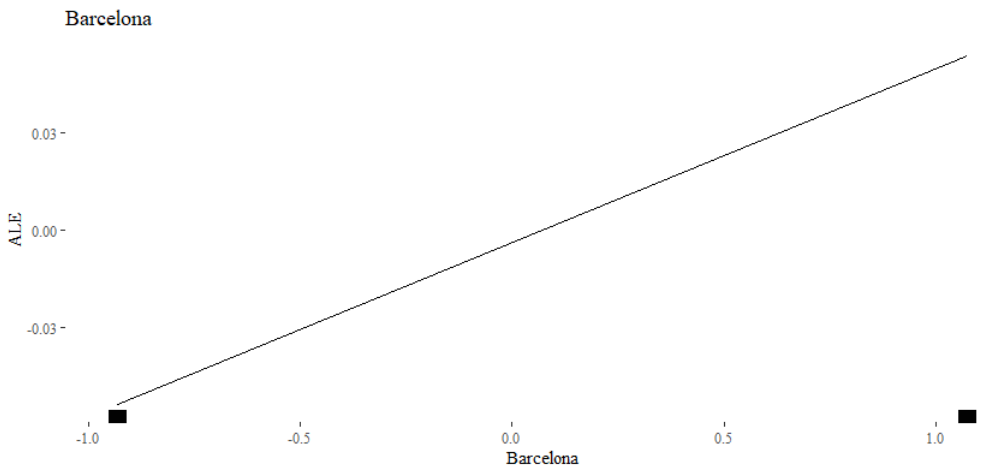


Figura 23: ALE Plot *Barcelona*

- Cuando la variable toma valor 0 la estimación del precio del alquiler es → $e^{log(price)-0{,}0522}$ = 1439,55€
- Cuando la variable toma valor 1 la estimación del precio del alquiler es → $e^{log(price)+0{,}0522}$ = 1597,97€

Categ.distr

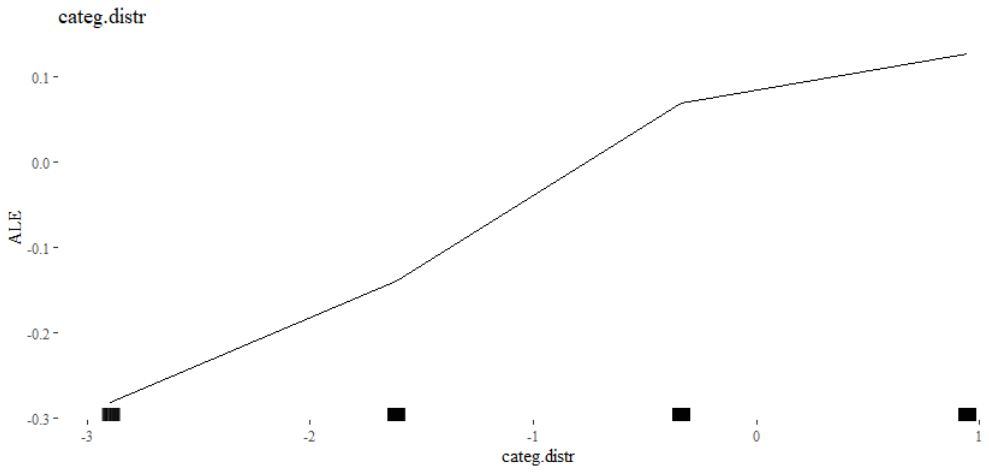


Figura 24: ALE Plot *categ.distr*

- Cuando la variable toma valor 1 la estimación del precio del alquiler es de 1105.33€
- Cuando la variable toma valor 2 la estimación del precio del alquiler es de 1368.49€
- Cuando la variable toma valor 3 la estimación del precio del alquiler es 1609.46€
- Cuando la variable toma valor 4 la estimación del precio del alquiler es 1701.07€

A diferencia del resultado obtenido mediante los gráficos de dependencia parcial en este caso si hay una variación significativa cuando la variable toma cada uno de los posibles valores. Tiene mucho sentido que sea de este modo porque cada nivel está calculado en base al precio medio del distrito en el que se encuentra la vivienda.

Type.chalet, type.duplex, type.penthouse y type.studio

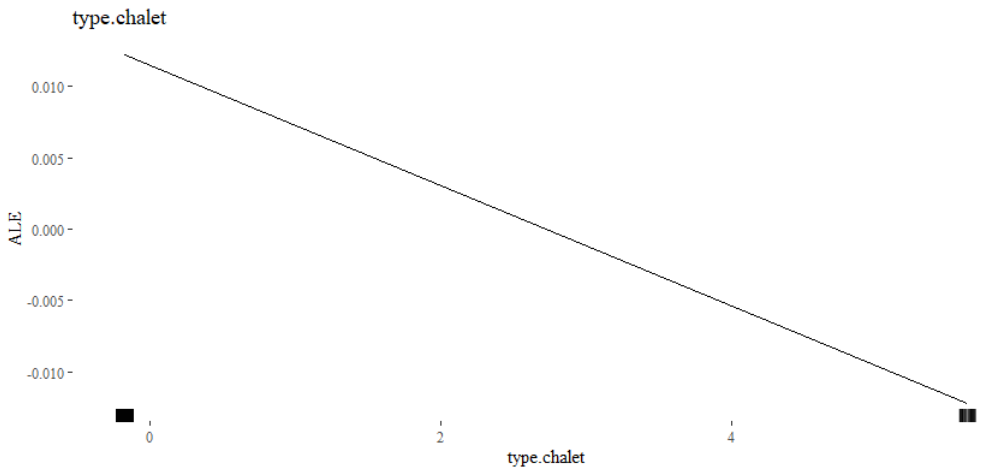


Figura 25: ALE Plot *type.chalet*

- Cuando la variable toma valor 0 la estimación del precio del alquiler es → $e^{log(price)+0{,}0158}$ = $1492{,}915€$
- Cuando la variable toma valor 1 la estimación del precio del alquiler es → $e^{log(price)-0{,}0158}$ = $1540{,}85€$

En este caso, la variación que produce la variable *type.chalet* es muy inferior a la estimada por los gráficos de dependencia parcial para la misma variable.

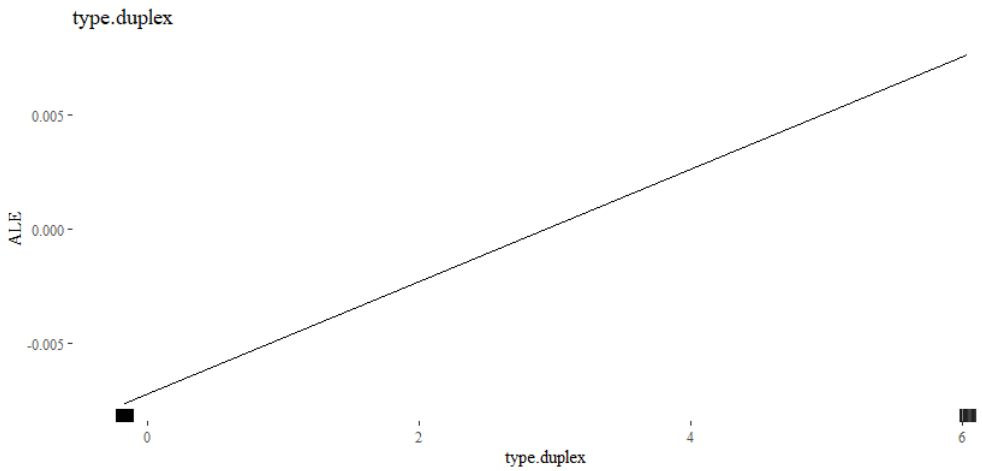


Figura 26: ALE Plot *type.duplex*

- Cuando la variable toma valor 0 la estimación del precio del alquiler es → $e^{log(price)-0{,}0067}$ = 1506,56€
- Cuando la variable toma valor 1 la estimación del precio del alquiler es → $e^{log(price)+0{,}0067}$ = 1526,89€

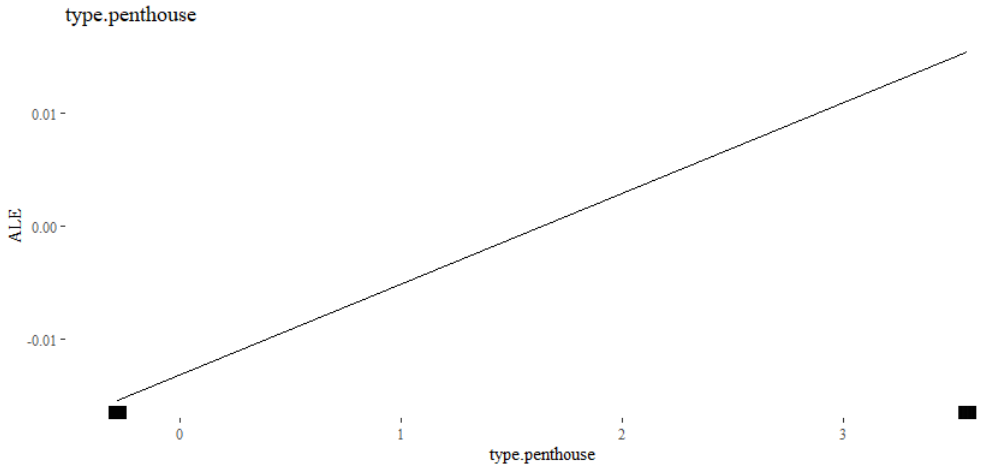


Figura 27: ALE Plot *type.penthouse*

- Cuando la variable toma valor 0 la estimación del precio del alquiler es → $e^{log(price)-0{,}0284}$ = 1474,22€
- Cuando la variable toma valor 1 la estimación del precio del alquiler es → $e^{log(price)+0{,}0284}$ = 1560,38€

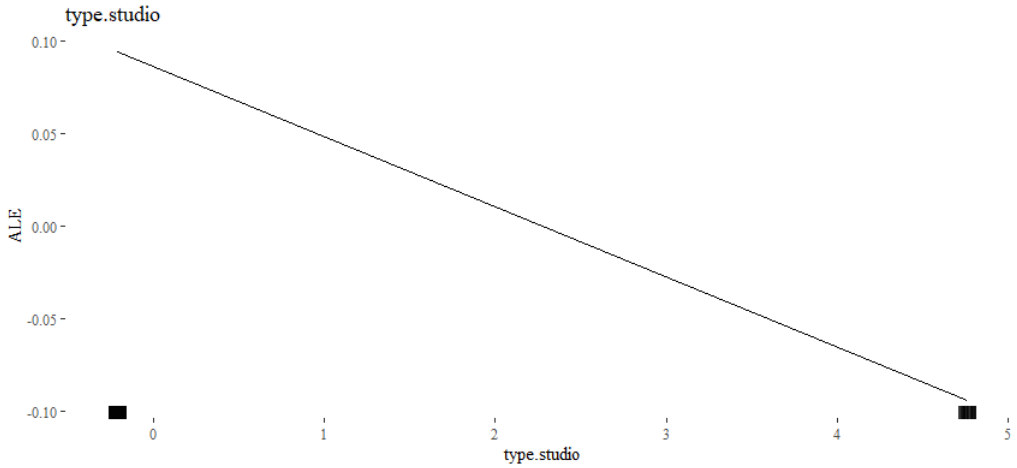


Figura 28: ALE Plot *type.studio*

- Cuando la variable toma valor 0 la estimación del precio del alquiler es → $e^{log(price)+0{,}0983} = 1673{,}35€$
- Cuando la variable toma valor 1 la estimación del precio del alquiler es → $e^{log(price)-0{,}0983} = 1374{,}69€$

A diferencia de los resultados obtenido en los gráficos de dependencia parcial en este caso la variable *type.chalet* tiene un efecto negativo en la estimación del precio de la vivienda, es posible que sea por una cuestión de localización de este tipo de vivienda. Por otro lado, el resto de variables tienen un efecto muy parecido aunque más suavizado.

Floor

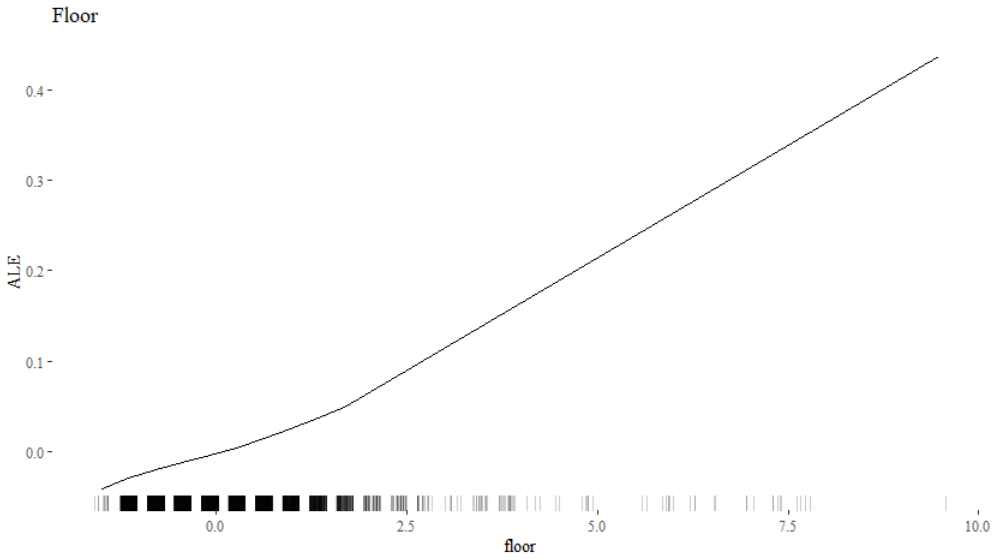


Figura 29: ALE Plot *floor*

- Cuando la variable toma su valor más pequeño la estimación del precio del alquiler es → $e^{log(price)-0{,}0408}$ = $1453{,}67€$
- Cuando la variable toma su valor más alto la estimación del precio del alquiler es → $e^{log(price)+0{,}4365}$ = $2342{,}87€$

Según el gráfico de efectos acumulados esta variable es bastante relevante ya que la variaicón en la estimación del precio en función de su valor es bastante grande y aumenta según aumenta el piso en el que se encuantra la vivienda.

hasLift

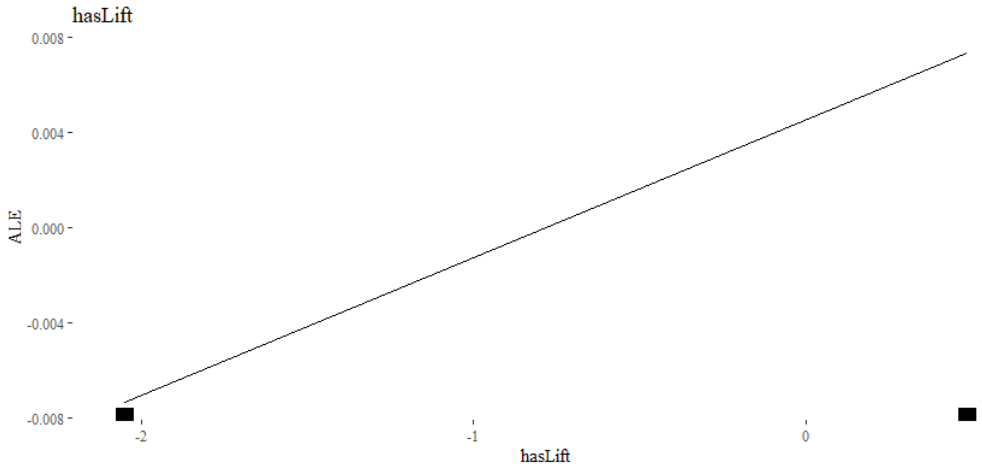


Figura 30: ALE Plot *hasLift*

- Cuando la variable toma valor 0 la estimación del precio del alquiler es → $e^{log(price)-0{,}0786}$ = $1402{,}04€$
- Cuando la variable toma valor 1 la estimación del precio del alquiler es → $e^{log(price)+0{,}0786}$ = $1640{,}71€$

En este caso, la variación en la estimación del precio del alquiler la vivienda es superior a la obtenida en los gráficos de dependencia parcial. Mediante este método el valor que toma esta variable es más relevante para el modelo.

FloorLift

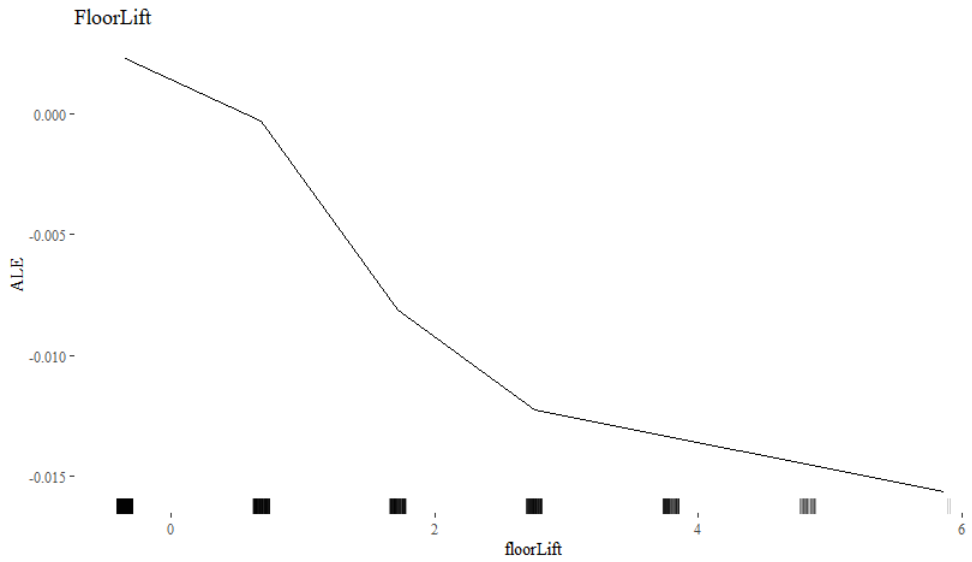


Figura 31: ALE Plot *floorLift*

- Cuando la variable toma su valor más pequeño la estimación del precio del alquiler es → $e^{log(price)+0{,}0023}$ = $1517{,}59€$
- Cuando la variable toma su valor más alto la estimación del precio del alquiler es → $e^{log(price)-0{,}0156}$ = $1490{,}65€$

Teniendo en cuenta el resultado obtenido mediante este método la variable no parece ser especialmente relevante para la estimación del precio de la vivienda que realiza la red neuronal.

Log.size

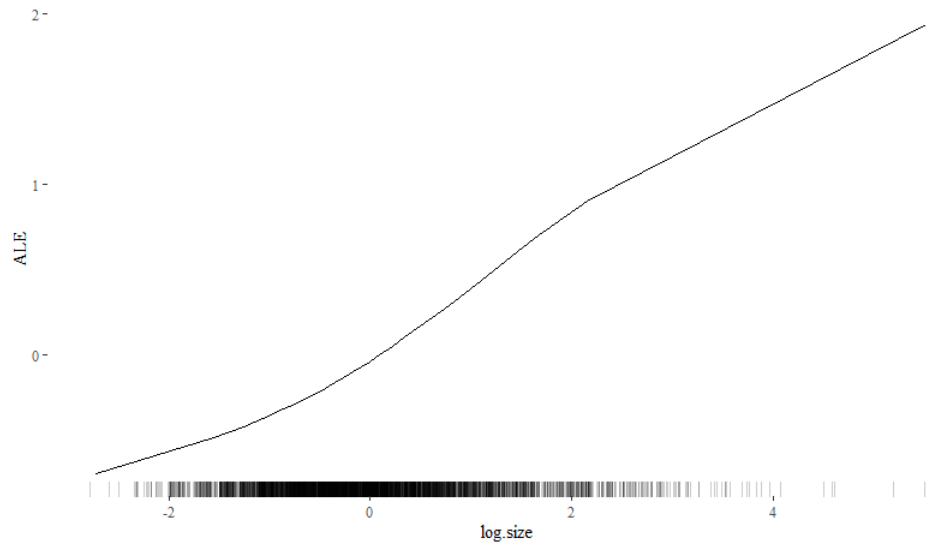


Figura 32: ALE Plot *log.size*

- Cuando la variable toma su valor más pequeño la estimación del precio del alquiler es → $e^{log(price)-0,7007}$ = $752,64€$
- Cuando la variable toma su valor más alto la estimación del precio del alquiler es → $e^{log(price)+1,9311}$ = $10460,76€$

La variable *log.size* vuelve a ser la que una mayor variación en el valor de la predición del precio del alquiler produce y tiene un rango muy similar al obtenido por los gráficos de dependencia parcial. Sin duda, es la variable con más relevancia para el modelo según los tres métodos empleados hasta el momento.

Exterior

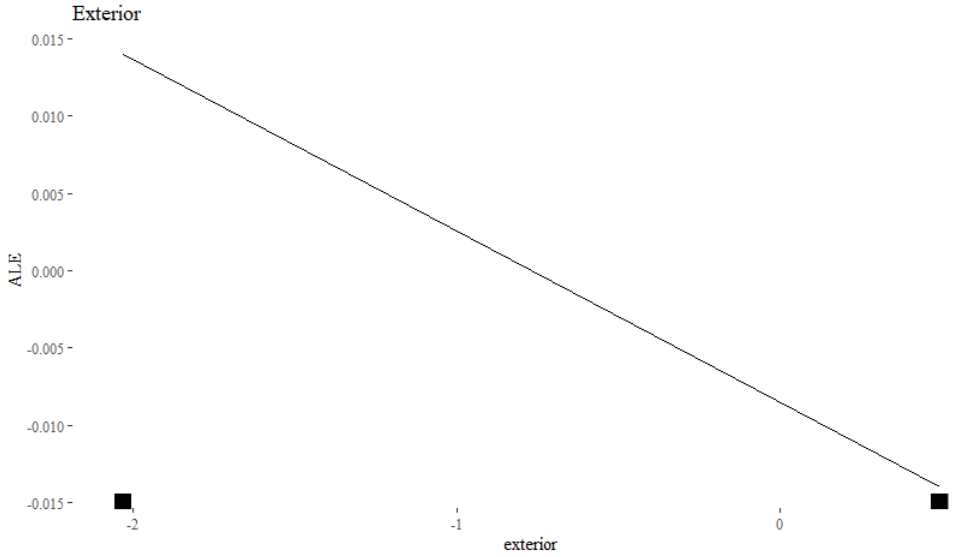


Figura 33: ALE Plot *exterior*

- Cuando la variable toma valor 0 la estimación del precio del alquiler es → $e^{log(price)-0,0139}$ = $1535,44€$
- Cuando la variable toma valor 1 la estimación del precio del alquiler es → $e^{log(price)+0,0139}$ = $1493,17€$

Este resultado es bastante inesperado ya que según el gráfico de efectos acumulados cuando una vivienda es exterior su precio esperado es inferior a cuando este no lo es.

Rooms

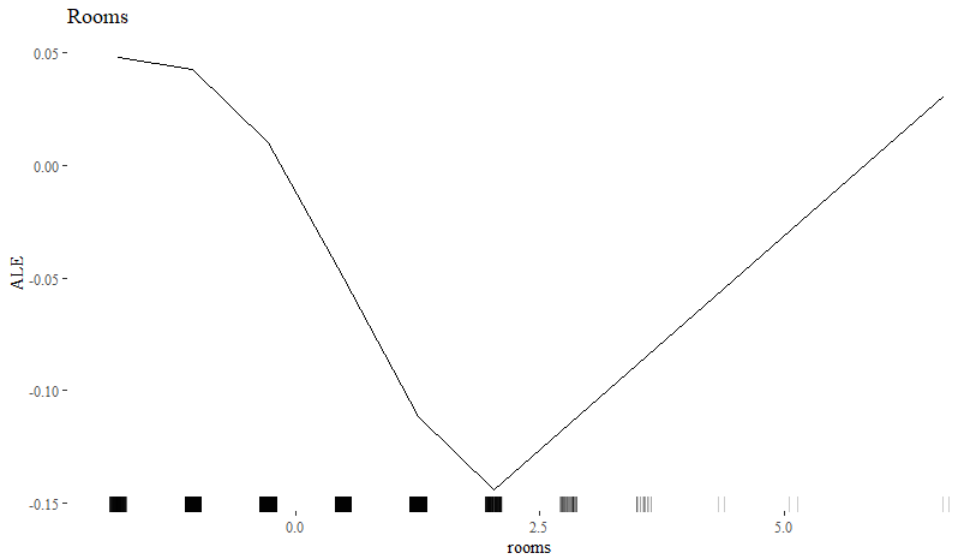


Figura 34: ALE Plot *rooms*

- Cuando la variable toma su valor más pequeño la estimación del precio del alquiler es → $e^{log(price)+0,0480}$ = 1562,07€
- Cuando la variable toma su valor medio la estimación del precio del alquiler es → $e^{log(price)-0,0499}$ = 1440,39€
- Cuando la variable toma su valor más alto la estimación del precio del alquiler es → $e^{log(price)+0,0312}$ = 1562,07€

Según el resultado obtenido podemos decir que según este método cuando la variable *rooms* toma valores de 0 a 4 el valor de la estimación del precio del alquiler decrece con cada habitación mientras que cuando la variable toma valor 5 o mayor la estimación del precio se vuelve a incrementar.

Bathrooms

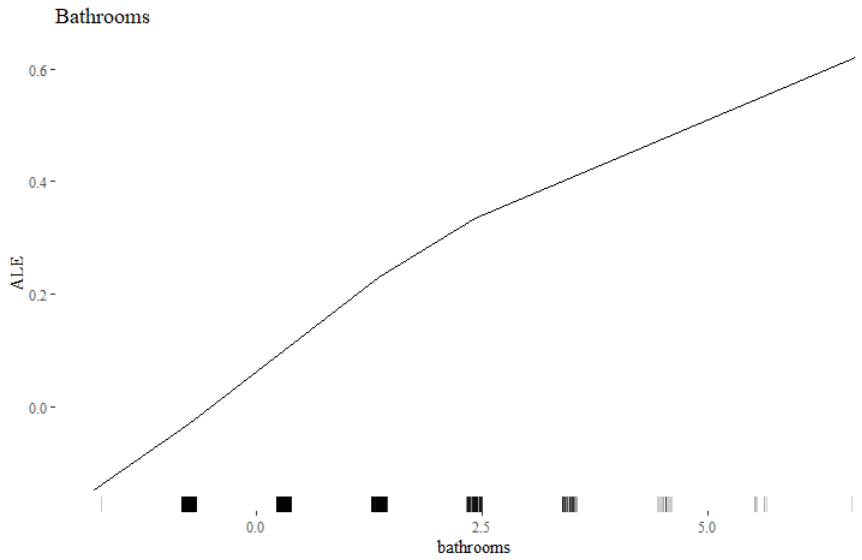


Figura 35: ALE Plot *bathrooms*

- Cuando la variable toma su valor más pequeño la estimación del precio del alquiler es → $e^{log(price)-0{,}1731}$ = $1273{,}41€$
- Cuando la variable toma su valor más alto la estimación del precio del alquiler es → $e^{log(price)+0{,}0312}$ = $3049{,}41€$

En este caso el número de baños que tiene la vivienda sí tiene una clara relevancia en la estimación del precio del alquiler de esta.

HasParkingSpace

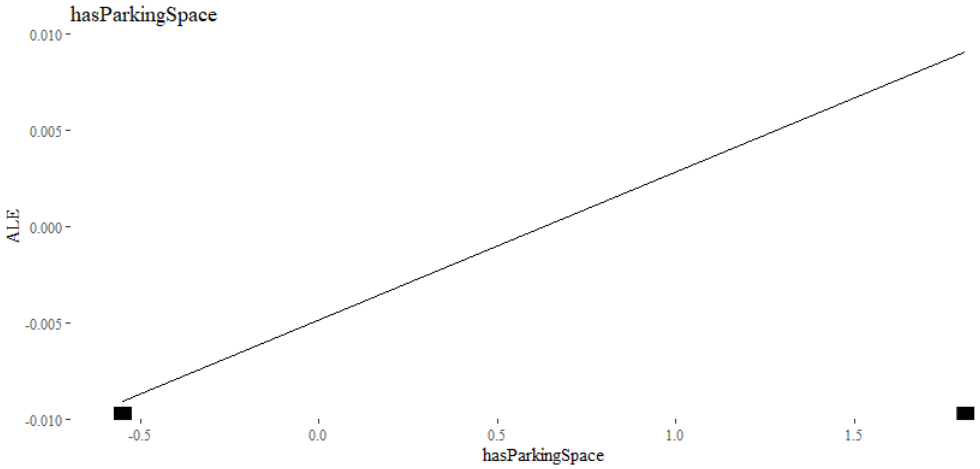


Figura 36: ALE Plot *hasParkingSpace*

- Cuando la variable toma valor 0 la estimación del precio del alquiler es → $e^{log(price)-0{,}0089}$ = $1503{,}25€$
- Cuando la variable toma valor 1 la estimación del precio del alquiler es → $e^{log(price)+0{,}0089}$ = $1530{,}25€$

En esta caso, el efecto de esta variable, al igual que según los métodos anteriores, no parece ser muy significativo en el cambio del valor estimado por el modelo para la variable *price*.

ParkingInPrice

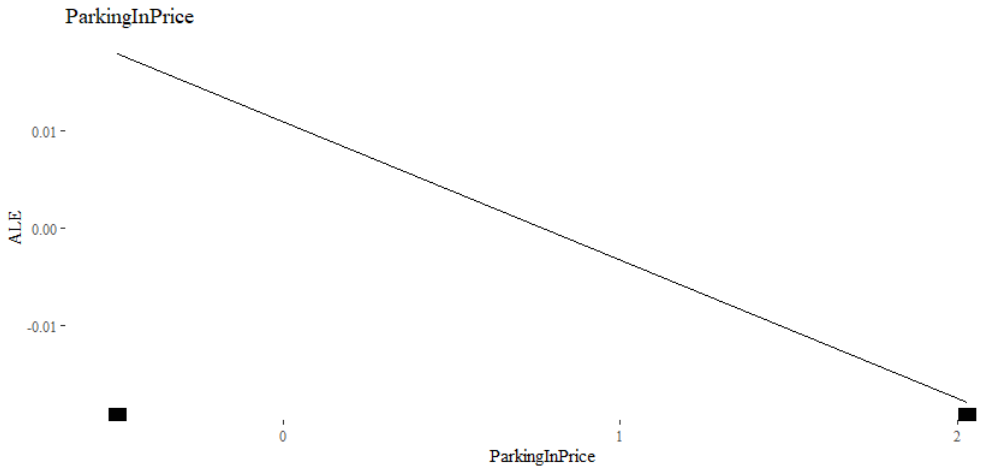


Figura 37: ALE Plot *ParkingInPrice*

- Cuando la variable toma valor 0 la estimación del precio del alquiler es → $e^{log(price)+0{,}0183}$ = $1544{,}70€$
- Cuando la variable toma valor 1 la estimación del precio del alquiler es → $e^{log(price)-0{,}0183}$ = $1489{,}18€$

La variable *ParkingInPrice* vuelve a tener un efecto negativo en la estimación que realiza el modelo cuando toma valor 1, indicando que el las viviendas que incluyen la plaza de aparcamiento en el precio del alquiler suelen ser más baratas que las que no. Este resultado es muy poco intuitivo ya que lo que sin analizar los datos lo natural sería pensar en que se produce el efecto contrario.

Log_activation

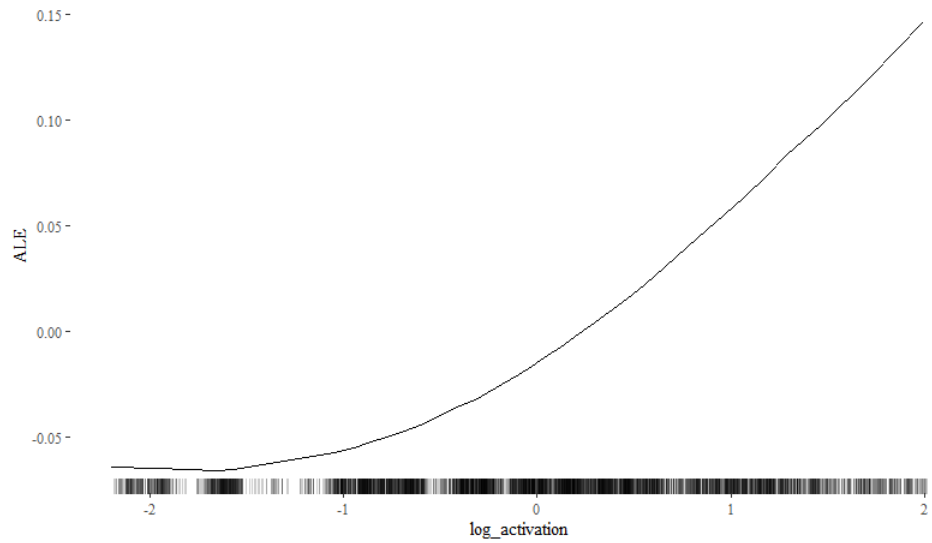


Figura 38: ALE Plot *log_activation*

- Cuando la variable toma su valor más pequeño la estimación del precio del alquiler es → $e^{log(price)-0{,}0644}$ = $752{,}64€$
- Cuando la variable toma su valor más alto la estimación del precio del alquiler es → $e^{log(price)+0{,}1464}$ = $1755{,}85€$

En este caso el valor la variable *log_activation* produce una variación bastante grande en la estimación del precio de la vivienda que realiza el modelo; podemos considerar esta variable bastante relevante según el método de efecto acumulados.

Resumen de resultados

Al igual que en los resultados obtenidos en los gráficos de dependencia parcial la variable que tiene una mayor relevancia según el método de los gráficos de efectos acumulados es *log.size*, seguida tambieén por *bathrooms* y *categ.distr*. En este caso, las variables *type.chalet* y *type.duplex* tienen un efecto mucho menor en la estimación del precio del alquiler de la que tienen según los gráficos de dependencia parcial. La variable *floorLift* vuelve a ser la menos relevante del conjunto de datos, seguida de *exterior* y *type.chalet*. En general, los resultados proporcionados por este método son bastante similares a los obtenidos por los gráficos de dependencia parcial pero con los rangos más suavizados.

4.2.4. Interacción de variables

Los gráficos que representan la fuerza de interacción de la variables de este capítulo se han realizado mediante la función `Interaction$new` del paquete *iml* de R. Tal y como se indica en el apartado explicativo de este método, lo habitual es que esta medida tome valores entre 0 y 1, por lo tanto consideraremos que una interacción es relevante cuando supere el 0.5 de fuerza de interacción.
En primer lugar, tenemos el gráfico de la fuerza de interacción global de todas las variables:

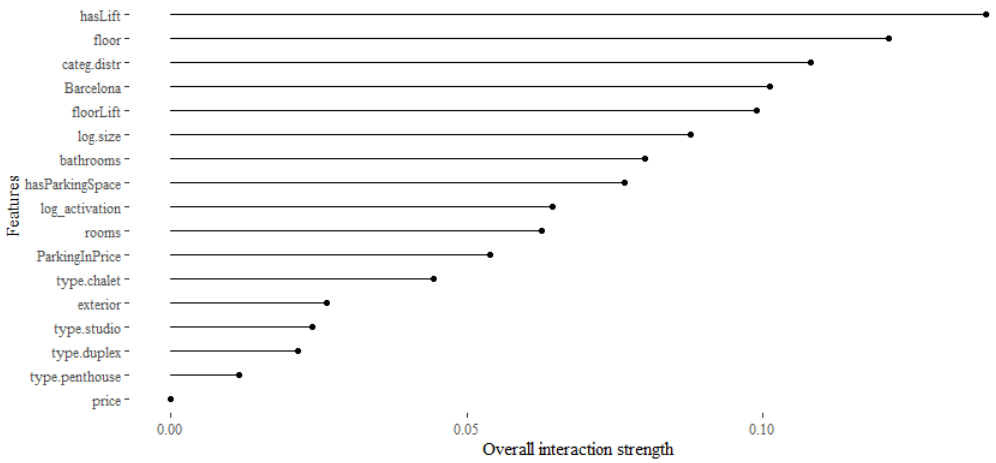


Figura 39: Gráfico de fuerza de interacción global

En primer lugar, resulta bastante llamativo que la variable *hasLift* sea la que mayor fuerza interacción presenta con el resto. Si bien es cierto que se espera que tenga una alta interacción con la variable *floorLift*, ya que está relacionada con el cálculo de ella, habrá que prestar atención en especial a su gráfico de fuerza de interacción individual para ver con qué otras variables está relacionada y estudiar si se puede tomar como una relevancia válida y no alterada por una alta correlación con otras variables.
Aun así, su fuerza de interacción es menor de los esperado ya que está por debajo de un 0.5 de fuerza en su interacción.

Barcelona

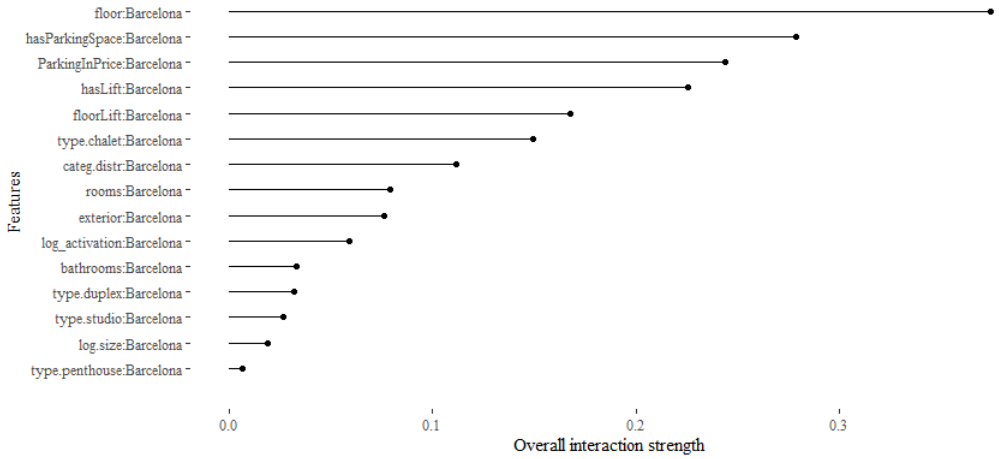


Figura 40: Gráfico de fuerza de interacción *Barcelona*

En el gráfico podemos observar como las variables con las que tiene una mayor interacción son *floor*, *hasParkingSpace*, *ParkingInPrice* y *hasLift*.
Sin embargo, ninguna de las variables tiene una fuerza de interacción especialmente alta (por debajo del 0.5).

Categ.distr

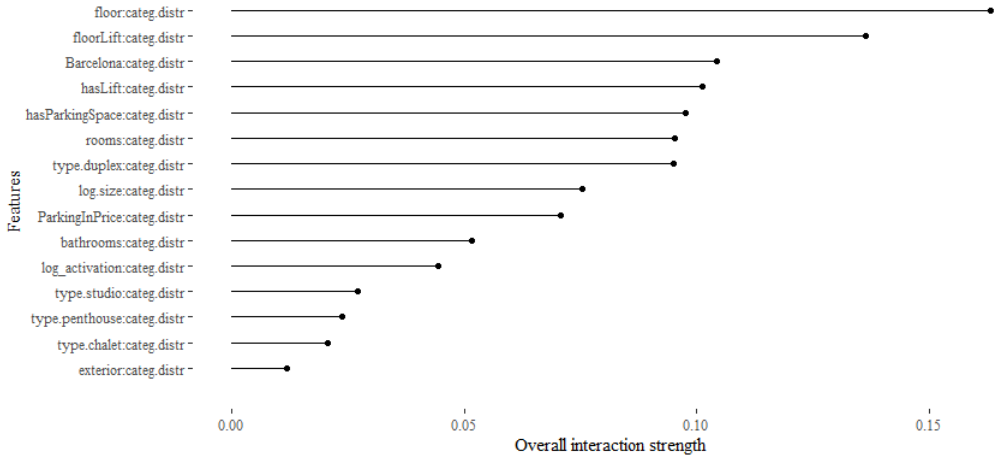


Figura 41: Gráfico de fuerza de interacción *categ.distr*

En este caso las interacciónes con el resto de variables no son tan fuertes como en otros casos. Aun así tres de las variables con las que tiene una mayor fuerza de interacción son *floor*, *floorLift* y *hasLift*; como estas tres variables sí están bastante correlacionadas es posible que este gráfico de importancia no sea del todo fiable para medir la fuerza de interacción de la variable.

Del mismo modo que en la variable anterior, ninguna de las variables tiene una fuerza de interacción especialmente alta; no aportan información relevante a la red.

Type.chalet

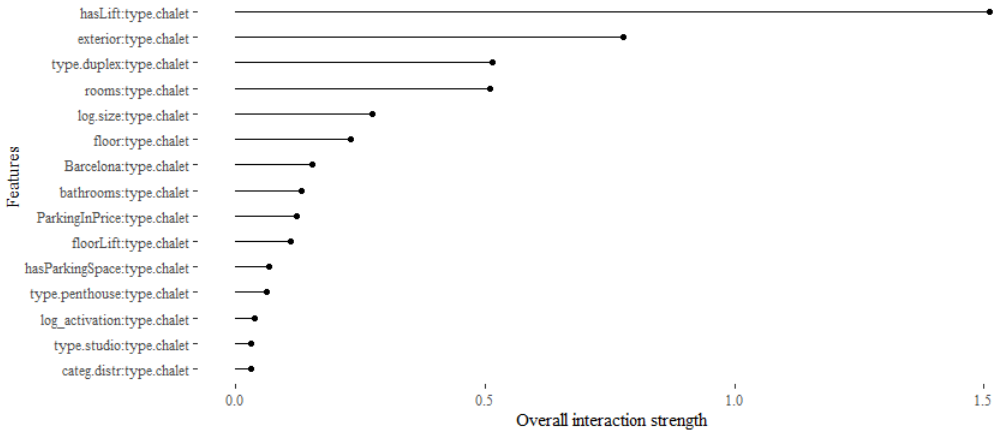


Figura 42: Gráfico de fuerza de interacción *type.chalet*

La variable *type.Chalet* tiene varias variables con una fuerza de interacción bastante alta, superior al 0.5. La variable con la que tiene una fuerza de interacción es *hasLift*, estas variables están ligeramente correlacionadas negativamente y tiene sentido que su fuerza de interacción sea alta.

Resulta bastante relevante que otra de las variables con las que tiene bastante fuerza de interacción es *type.duplex*.

Type.duplex

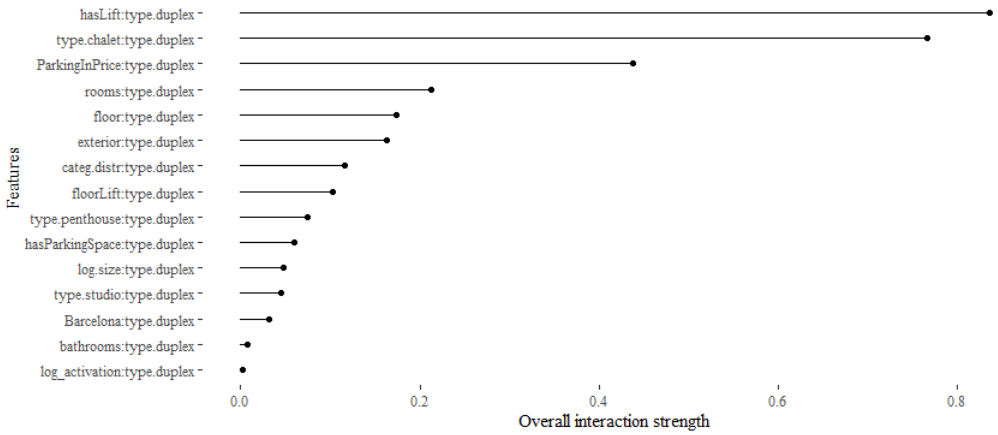


Figura 43: Gráfico de fuerza de interacción *type.duplex*

En este caso las variables que aportan una mayor fuerza de interacción son *hasLift* y *type.Chalet*.

Type.penthouse

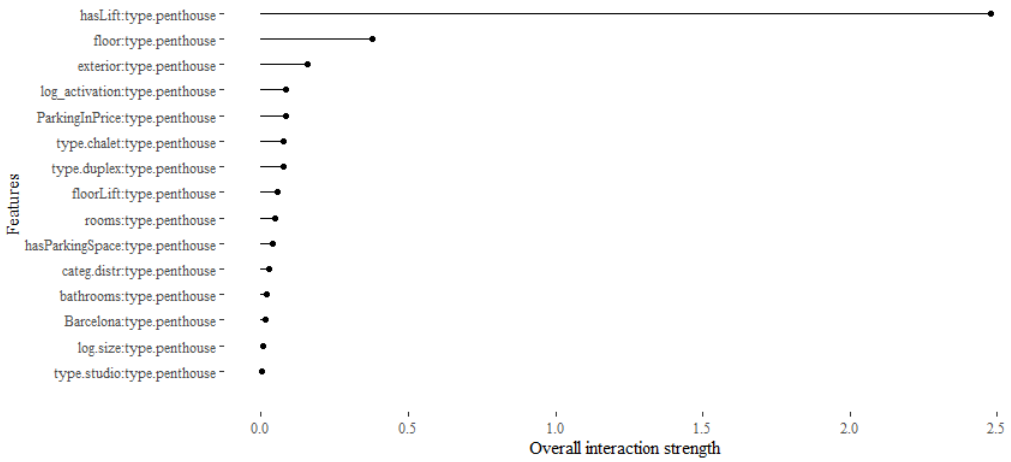


Figura 44: Gráfico de fuerza de interacción *type.penthouse*

La variable *type.penthouse* tiene una fuerza de interacción muy alta con la variable *hasLift*. La fuerza de esta interacción tiene bastante sentido ya que es habitual que un ático esté en un edificio con ascensor.

Type.studio

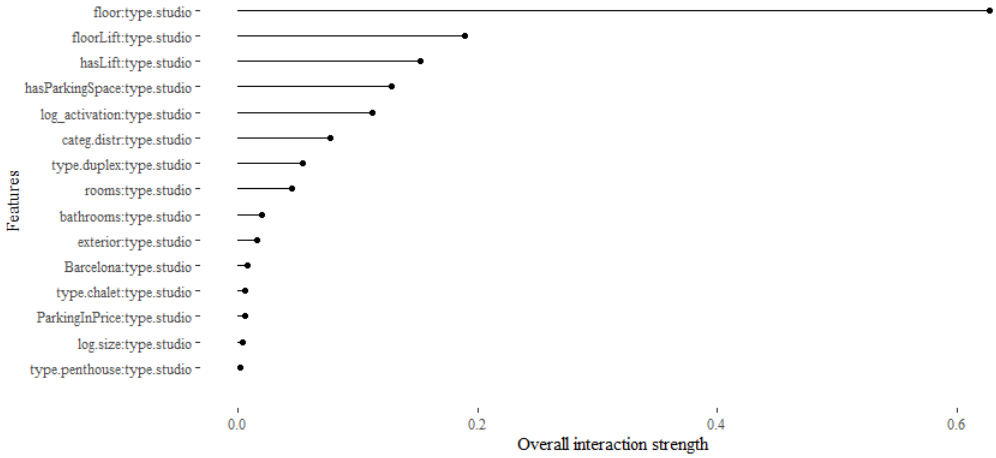


Figura 45: Gráfico de fuerza de interacción *type.studio*

En este caso, la variable *type.studio* tiene una mayor fuerza de interacción con la variable *floor*, su fuerza de interacción no es particularmente alta peo sí aporta información relevante a la red neuronal.

Floor

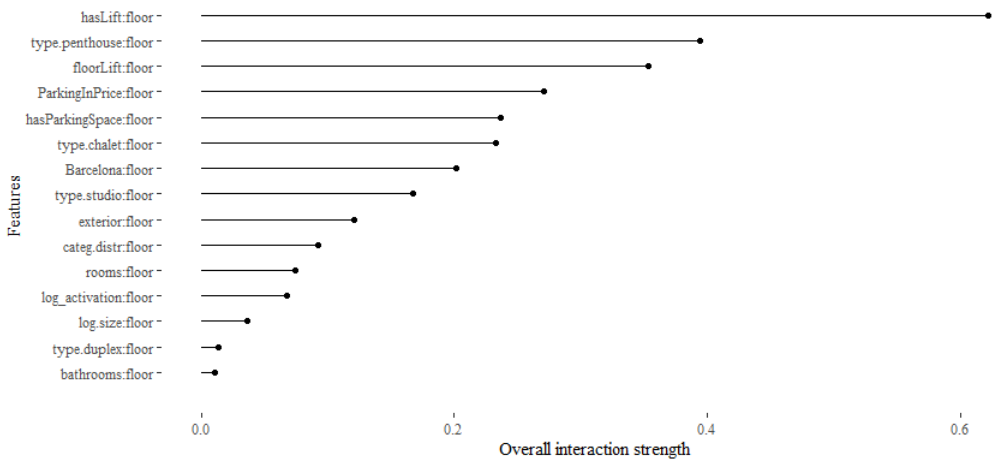


Figura 46: Gráfico de fuerza de interacción *floor*

La mayor fuerza de interacción de esta variable se da con *hasLift*. En el conjunto de datos ambas variables están ligeramente correlacionadas de forma positiva, sin embargo, su fuerza de interacción no es tan fuerte como se podría esperar.

FloorLift

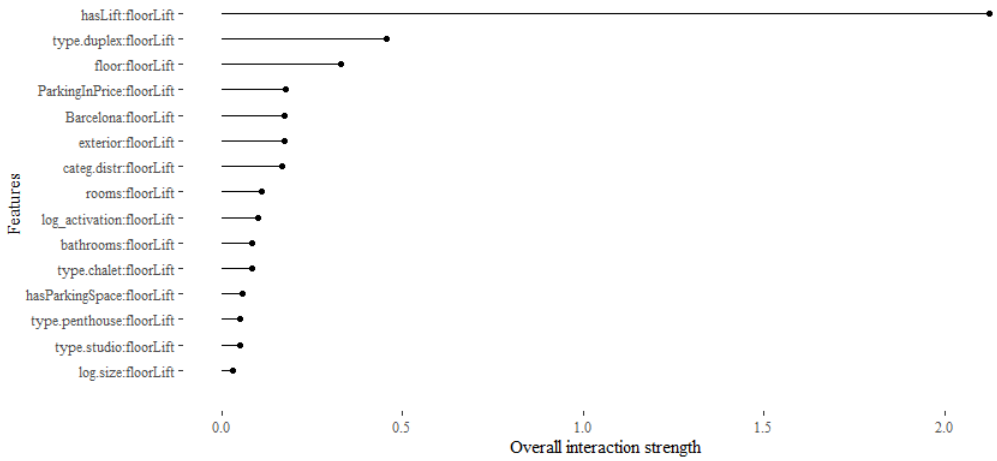


Figura 47: Gráfico de fuerza de interacción *floorLift*

En En este caso, la variable *floorLift* tiene una fuerza de interacción muy alta con la variable fuerte con *hasLift*. Teniendo en cuenta que *floorLift* es una variable calculada a partir de las variables *floor* y *hasLift* no podemos definir con claridad si realmente esta interacción aporta información relevante a la red neuronal.

Log.size

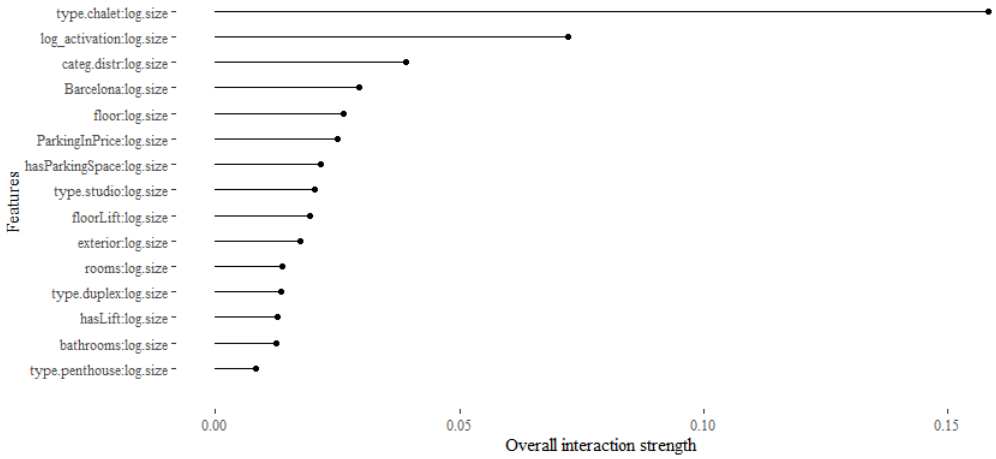


Figura 48: Gráfico de fuerza de interacción *log.size*

La variable *log.size* tiene claramente su mayor fuerza de interacción con la variable *type.chalet*; sin embargo su fuerza no es demasiado alta. Podemos asumir que ninguna interacción de la variable *log.size* con otra del modelo aporta información muy valiosa a la red neuronal.

Exterior

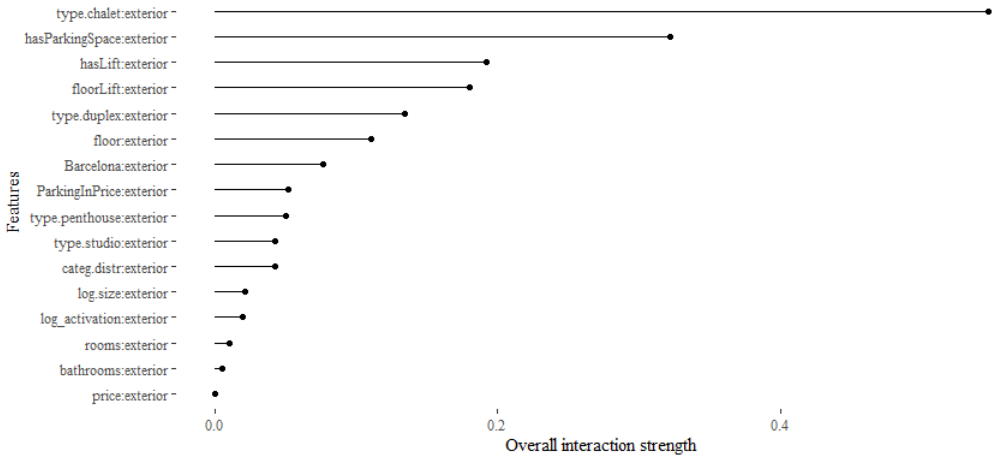


Figura 49: Gráfico de fuerza de interacción *exterior*

En este caso, la variable *exterior* tiene su mayor fuerza de interacción con la variable *type.chalet*; tal y como hemos visto antes para esta otra variable. Este nivel de fuerza tiene bastante sentido ya que todos los chalets son exteriores.

Rooms

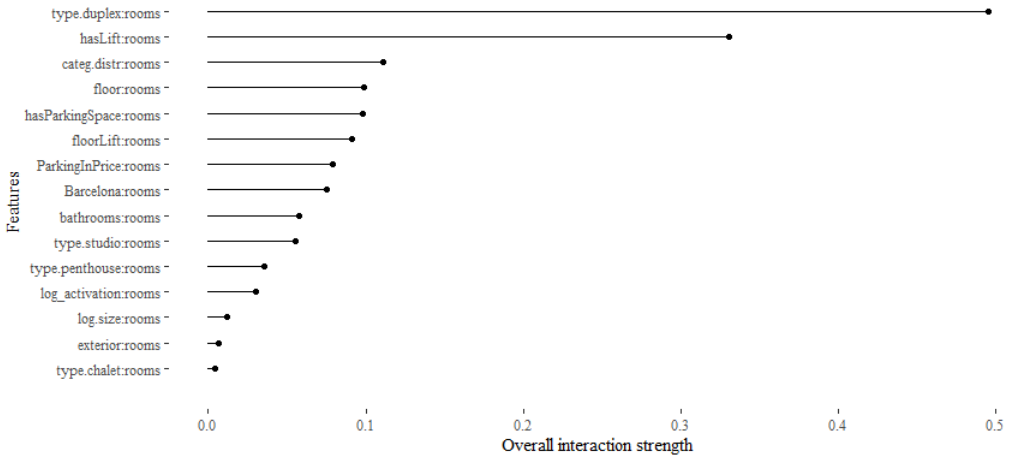


Figura 50: Gráfico de fuerza de interacción *rooms*

La variable *rooms* tiene su mayor fuerza de interacción con la variable *type.duplex*, sin embargo, tal y como pasa en otras variables su fuerza de interacción no es lo suficientemente grande como para considerarla relevante.

Bathrooms

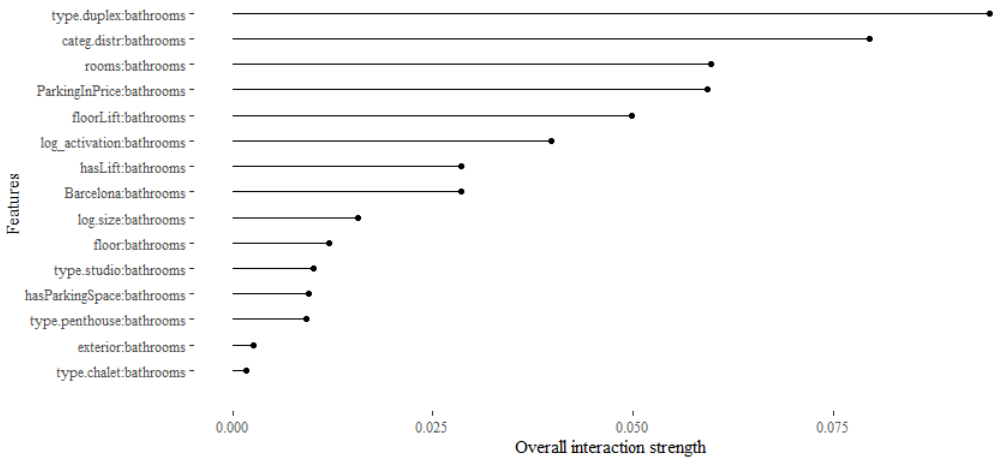


Figura 51: Gráfico de fuerza de interacción *bathrooms*

En este caso ninguna interacción con la variable *barhroom* supera el 0.1 de fuerza de interacción. Ninguna de sus interaciones aporta valor a la red neuronal.

HasParkingSpace

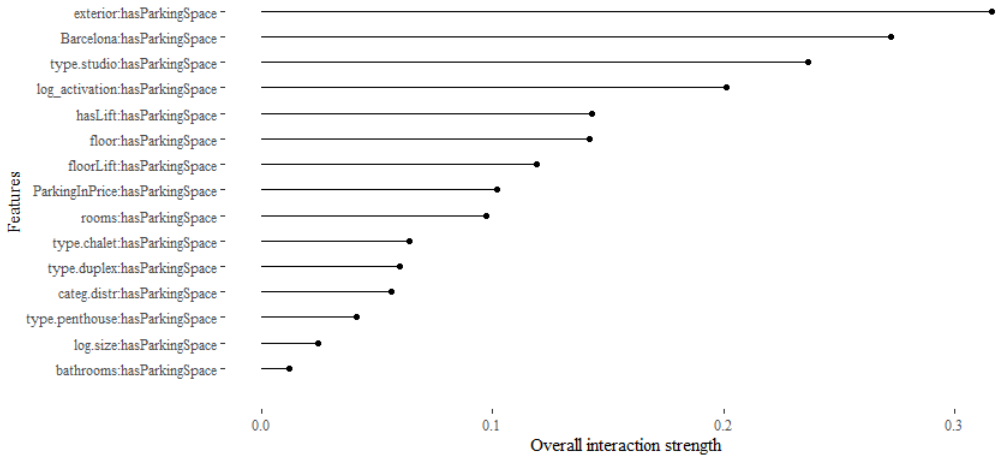


Figura 52: Gráfico de fuerza de interacción *hasParkingSpace*

La variable *hasParkingSpace* no tiene una gran fuerza de interacción con ninguna otra variable, Aunque su mayor fuerza se da con la variable *exterior* no es lo suficientemente grande como para que se considere relevante.

ParkingInPrice

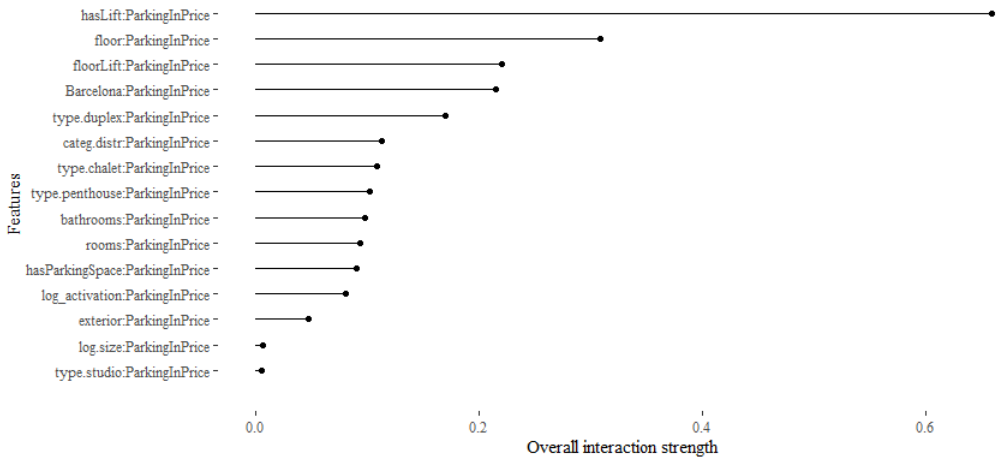


Figura 53: Gráfico de fuerza de interacción *ParkingInPrice*

En este caso la variable *ParkingInPrice* tiene su mayor fuerza de interacción con la variable *hasLift*, sin embargo, su fuerza de interacción no es demasiado grande y no queda claro si realmente aporta información valiosa al modelo.

Log.activation

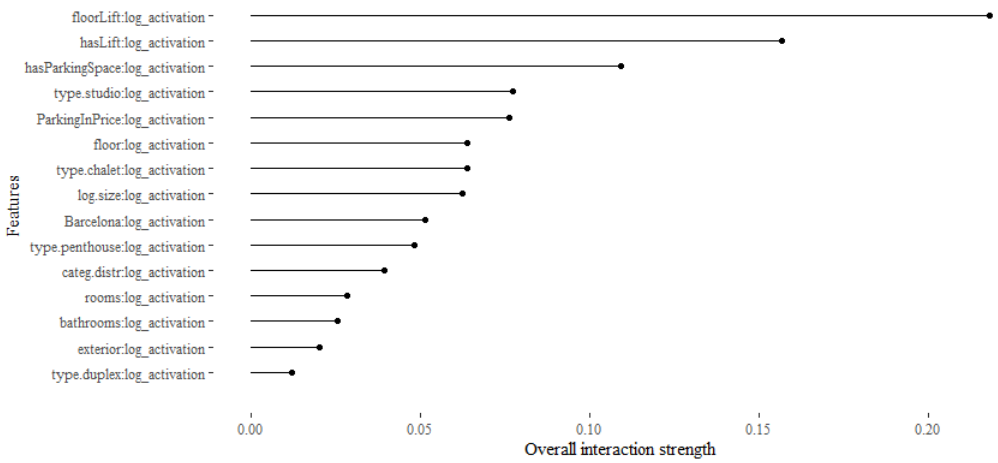


Figura 54: Gráfico de fuerza de interacción *log_activation*

La variable *log_activation* tiene su mayor fuerza de interacción con la variable *floorLift* pero al igual que en los casos anteriores su fuerza de interacción es muy poco significativa y no podemos considerar que aporte información de valor a la red neuronal.

4.2.5. Método de las permutaciones

El método de las permutaciones tiene una implementación muy sencilla en R ya que existen varios paquetes que permiten que se utilice muy fácilmente. En este caso se ha utilizado el paquete *vif* para calcular las importancias y hacer un gráfico con el resultado.

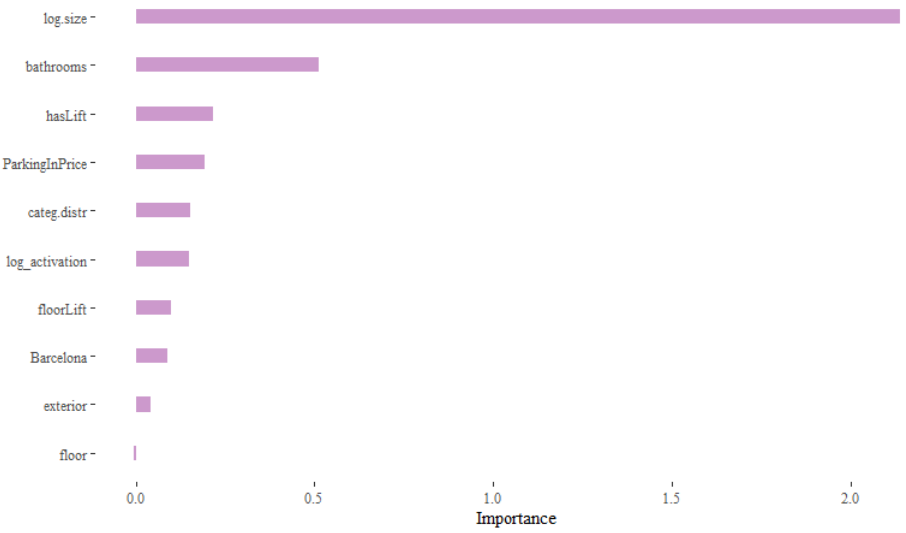


Figura 55: Relevancia mediante el método de las permutaciones

Mediante la utilización de este método observamos que las variables con una mayor importancia en la predición de la red neuronal son *log.size*, *type.duplex* y *bathrooms*. Mientras que

4.3. Método de las variables fantasma

La implementación para realizar el cálculo de la relevancia mediante el uso de variables fantasma ha sido más complicada que para el resto de medidas debido a que por el momento no está documentada en ningún paquete de R. El código empleado para realizar los cálculos se puede encontrar en el *Anexo II* de esta memoria. Los resultados obtenidos han sido:

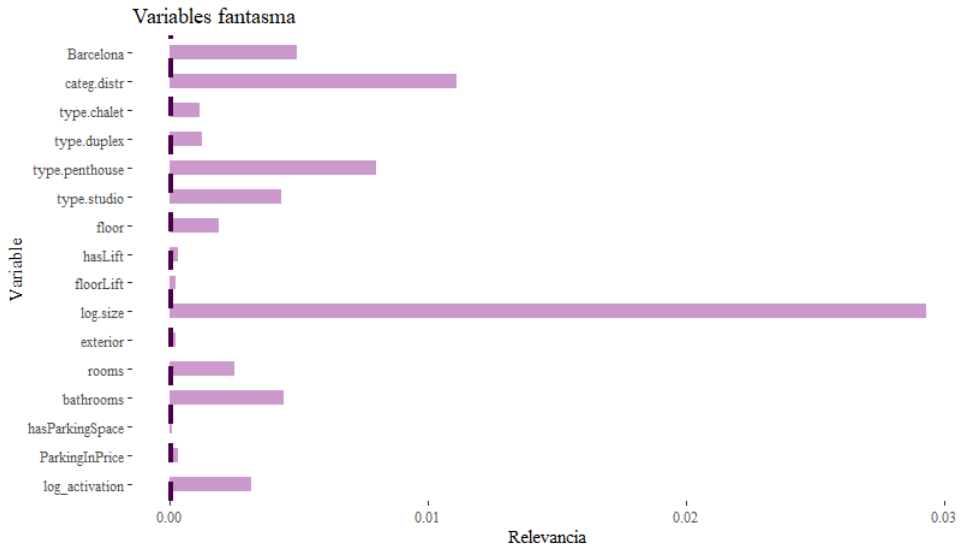


Figura 56: Relevancia mediante variables fantasma

Al igual que en en el resto de métodos la variable con una mayor relevancia ha sido *log.size* seguida de la variable *categ.distr*. Los resultados también coinciden respecto a la variables que son menos relevantes como *hasParkingSpace*.

5. Conclusiones

Tras concluir el proceso de investigación y análisis de este trabajo es posible realizar una valoración general sobre cada uno de los métodos y lo que aporta cada uno de ellos a una mejor interpretación de las variables explicativas que utiliza una red neuronal. Antes de presentar las conclusiones obtenidas, es importante recordar que los objetivos de este trabajo han sido conocer los métodos empleados para analizar la importancia que tiene una variable en el *output* que calcula una red neuronal y compararlos de una forma práctica con el método planteado en el artículo *Variable relevance by ghost variables* (Delicado and Peña 2019).
Tanto en la parte teórica como en la parte práctica de este trabajo se han alcanzado la mayoría de objetivos propuestos al inicio de la investigación. En lo referente a la parte teórica hemos logrado definir cuáles son los métodos empleados actualmente para analizar la relevancia de una variable en un red neuronal y qué diferencias hay entre ellos, el proceso de recopilación de información resultó particularmente largo ya que había que comprender bien cómo funciona una red neuronal y además la literatura sobre su interpretación que, aunque está en crecimiento, no es demasiado extensa. En lo referente a la parte práctica, el análisis ha resultado bastante complicado, no solo por el proceso de entrenar una red neuronal sino también por identificar qué paquetes de R eran necesarios para implementar cada método y como debía utilizarse cada uno de ellos. Finalmente, hemos podido comprobar cuáles son las diferencias en la interpretación de cada uno de los diferentes métodos y también hemos descubierto que no todos ellos aportan el mismo tipo de información sobre las variables analizadas. Las conclusiones obtenidas más se exponen a continuación.

En primer lugar, ningún método proporciona un análisis completo sobre la relevancia de una variable por sí solo. Para poder saber cómo de relevante es una variable de un conjunto de datos mediante los métodos gráficos descritos en este trabajo es necesario dibujar todas la variables para poder comparar el efecto que tienen en la variable respuesta y poder determinar si son relevantes o no. Por otro lado, los métodos que solo calculan un valor para la relevancia de la variable no permiten saber qué tipo de efecto tienen sobre la variable respuesta, a diferencia de los métodos gráficos, que permiten conocer qué valores incrementan o disminuyen la estimación de la variable respuesta.
Respecto al método que calcula la fuerza de las interacciones entre las variables y su relevancia, es muy útil para definir qué interacciones son relevantes pero no para estudiar la relevancia de una variable por sí misma. El análisis de la relevancia de la interacciones de las variables no la ofrece ningún otro método.
En lo referente a los métodos gráficos, es cierto que los gráficos de efectos acumulados son igual de eficientes que los gráficos de dependencia parcial y menos sensibles a los errores por correlación entre variables pero la elección del método dependerá de la persona que vaya a realizar el análisis ya que los gráficos de dependencia parcial son mucho más sencillos de interpretar.

Finalmente, en lo referente a los métodos que proporcionan una única medida de relevancia numérica para cada una de las variables el método de calculo de relevancia mediante la omisión

de variables es el más ineficiente de todos, no solo es muy lento de computar sino que además pierde mucha información de valor como puede ser la relevancia de una interacción para el modelo. En lo que respecta a los métodos de relevancia mediante permutaciones y variables fantasma, este último es mucho menos sensible a errores debido a la correlación entre variables pero actualmente su implementación mediante código es mucho más compleja. Actualmente se está trabajando en una librería de R que facilite la utilización de este método.

Para cerrar esta memoria, me gustaría recordar que hay algunos aspectos en la aplicación del método de variables fantasma que deberían analizarse más a fondo, como por ejemplo si es posible extender su uso a problemas de clasificación y no de regresión. Para esto es fundamental continuar con su desarrollo teórico con el objetivo de lograr un método versátil y robusto frente a los problemas que presentan el resto de métodos comentados en este trabajo.

Referencias

Delicado, P. and D. Peña (2019, dec). Understanding complex predictive models with Ghost Variables.

Fisher, A., C. Rudin, and F. Dominici (2018, 01). Model class reliance: Variable importance measures for any machine learning model class, from the rashomon"perspective.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. Technical Report 5.

Molnar, C., B. Bischl, and G. Casalicchio (2018). iml: An r package for interpretable machine learning. *JOSS@*(26), 786.

Anexo I

Código R

```

data("Boston", package = "MASS")
library(randomForest)
library(iml)
library(ggplot2)
library(ggthemes)
library(gridExtra)

rf <- randomForest(medv ~ ., data = Boston, ntree = 50)
mod <- Predictor$new(rf, data = Boston)


## PDP ##

eff <- FeatureEffect$new(mod,
feature = "rm", method = "pdp",
grid.size = 30
)

p1<-plot(eff) +
ggtitle("Dependencia parcial rm")+theme_tufte()

eff$set.feature("black")


p2<-plot(eff) +
ggtitle("Dependencia parcial black") +theme_tufte()+ylab("")


eff$set.feature("lstat")

p3<-plot(eff) +
ggtitle("Dependencia parcial lstat") +theme_tufte()+ylab("")

grid.arrange(p1, p2, p3, ncol=3)



## ALE ##


eff <- FeatureEffect$new(mod,
feature = "rm", method = "ale",
grid.size = 30
)

```

```
p1<-plot(eff) +
ggtitle("Efectos acumulados rm")+theme_tufte()

eff$set.feature("black")


p2<-plot(eff) +
ggtitle("Efectos acumulados black") +theme_tufte()+ylab("")


eff$set.feature("lstat")

p3<-plot(eff) +
ggtitle("Efectos acumulados lstat") +theme_tufte()+ylab("")

grid.arrange(p1, p2, p3, ncol=3)


## Interaciones ##

interact <- Interaction$new(mod)
plot(interact)+theme_tufte()


interact <- Interaction$new(mod, feature = "rm")
p1<-plot(interact)+theme_tufte()

interact <- Interaction$new(mod, feature = "black")
p2<-plot(interact)+theme_tufte()

interact <- Interaction$new(mod, feature = "lstat")
p3<-plot(interact)+theme_tufte()

grid.arrange(p1, p2, p3, ncol=3)


## Permutaciones ##

vip(rf, fill="#800080", alpha=.4, width=.4)+theme_tufte()
```

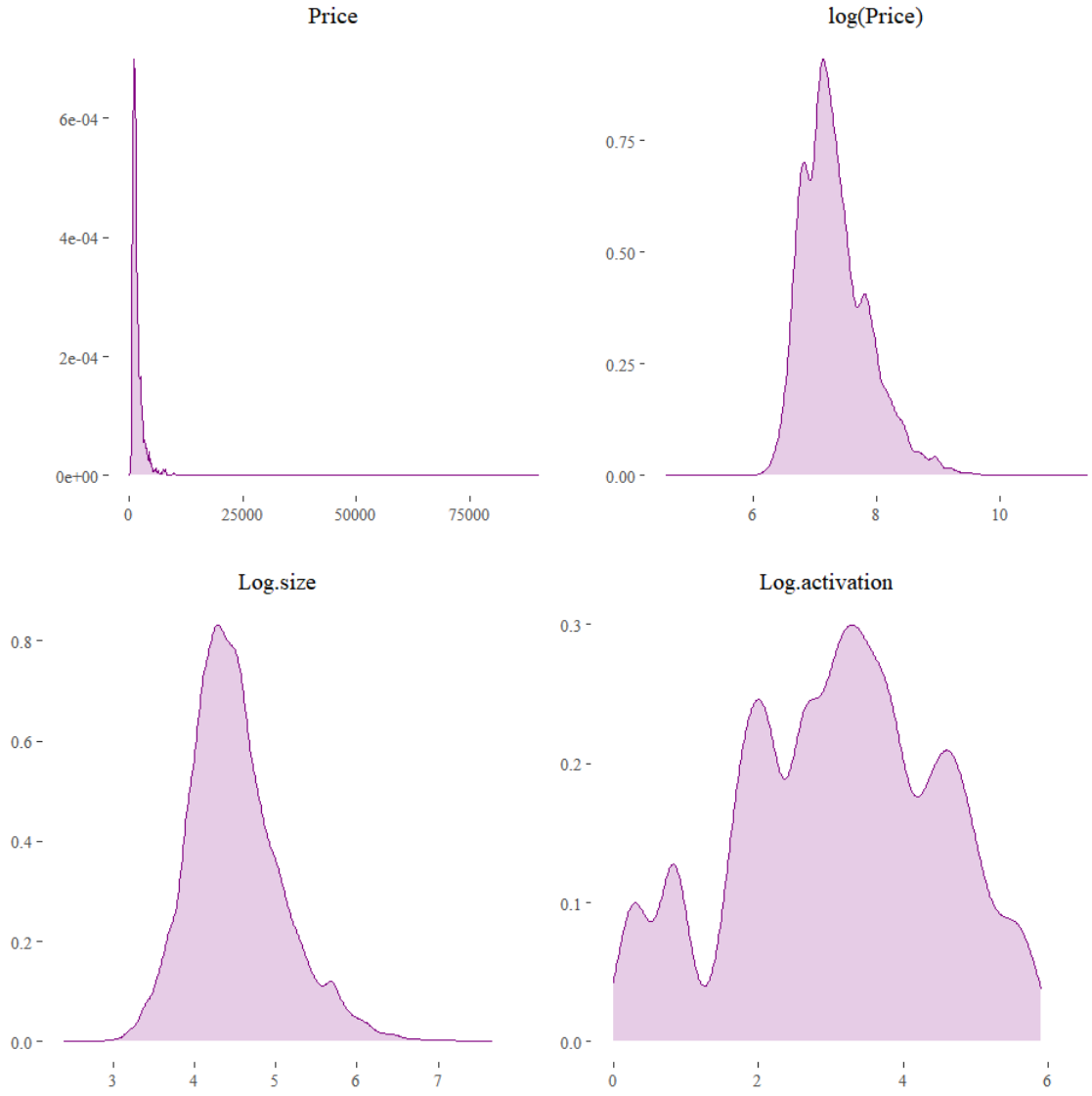
Anexo II

Análisis exploratorio de los datos

En este apartado podemos encontrar un pequeño análisis descriptivo del conjunto de datos *Idealista*.

Variables numéricas continuas

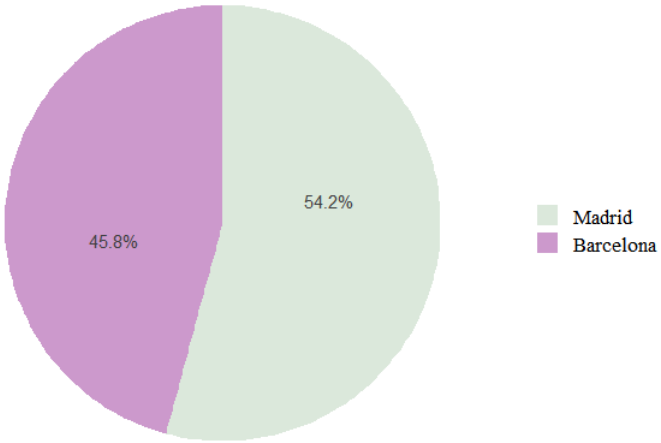
	Mean	Sd	Median	Min	Max	Range	Skew	Kurtosis	Se
price	1824.84	1580.31	1400.00	100.0	90000.00	89900.00	13.83	612.65	12.31
log.size	4.51	0.57	4.44	2.4	7.66	5.27	0.77	1.21	0.00
log_activation	3.09	1.39	3.15	0.0	5.90	5.90	-0.20	-0.61	0.01



Variables numéricas discretas y variables categóricas

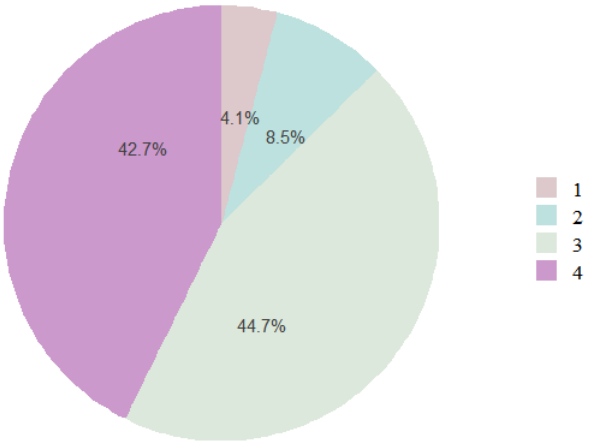
Barcelona

	F. absoluta	%
Madrid	8934	54.2
Barcelona	7546	45.8
TOTAL	16480	100.0



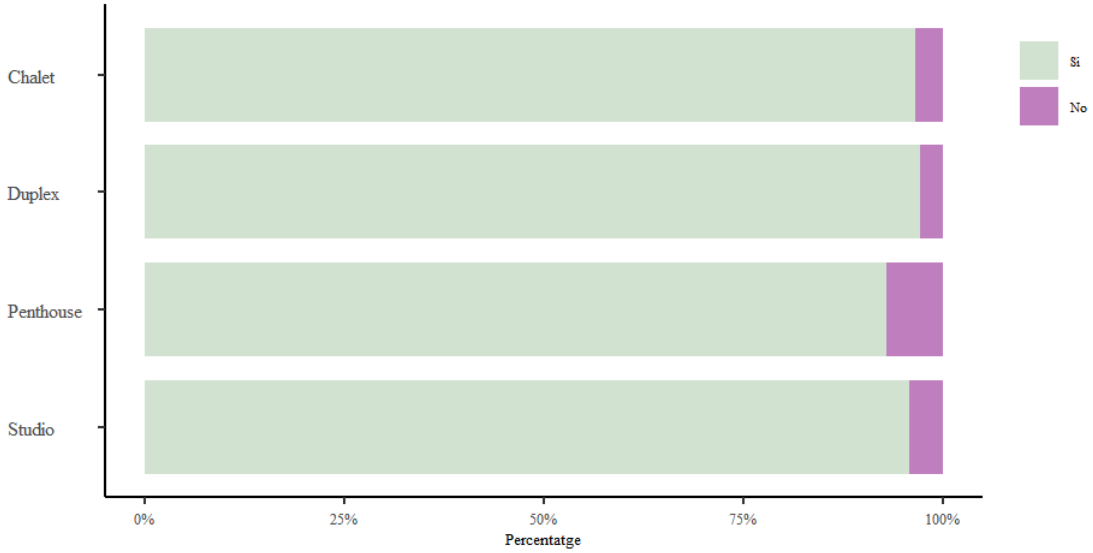
categ.distr

	F. absoluta	%
1	675	4.1
2	1397	8.5
3	7366	44.7
4	7042	42.7
TOTAL	16480	100.0



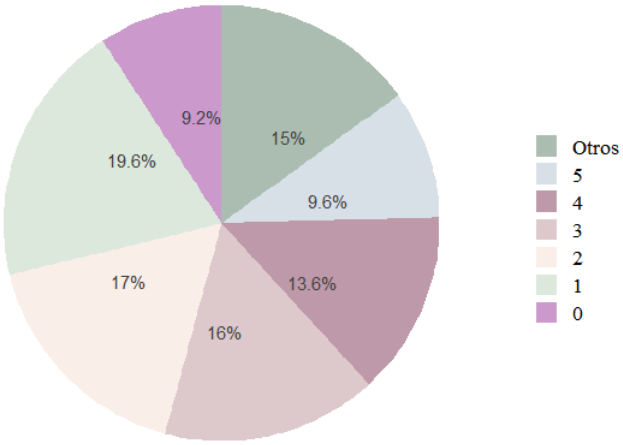
Type.chalet, type.duplex, type.penthouse y type.studio

	No		Sí		
	F. absoluta	**%**	**F. absoluta**	**%**	**TOTAL**
Chalet	15922	96.61	558	3.39	16480
Duplex	16019	97.20	461	2.80	16480
Penthouse	15312	92.91	1168	7.09	16480
Studio	15794	95.84	686	4.16	16480



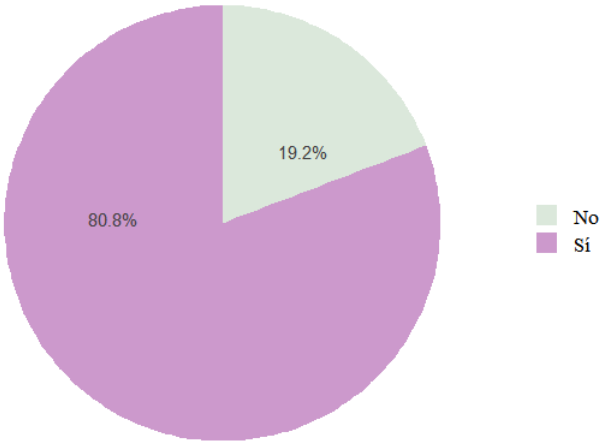
Floor

Piso	F. absoluta	%
-1	36	0.22
0	1524	9.25
1	3233	19.62
2	2800	16.99
3	2645	16.05
4	2248	13.64
5	1574	9.55
6	960	5.83
7	591	3.59
8	315	1.91
9	155	0.94
10	107	0.65
11	55	0.33
12	33	0.20
13	34	0.21
14	46	0.28
15	17	0.10
16	7	0.04
17	21	0.13
18	2	0.01
19	4	0.02
20	22	0.13
21	13	0.08
22	7	0.04
23	4	0.02
24	5	0.03
25	16	0.10
26	2	0.01
30	1	0.01
31	2	0.01
35	1	0.01
TOTAL	16480	100.00



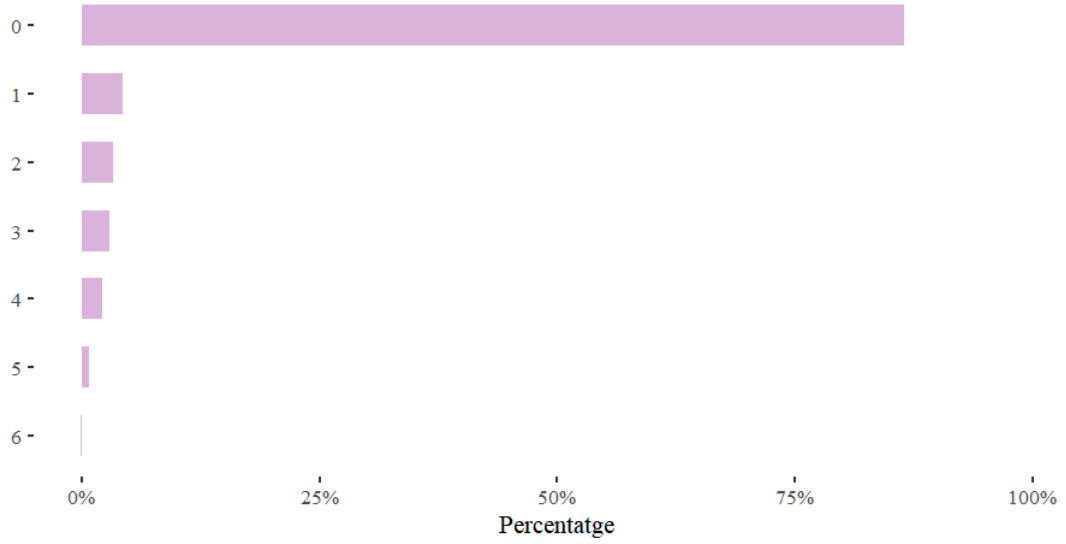
HasLift

	F. absoluta	%
No	3167	19.2
Sí	13313	80.8
TOTAL	16480	100.0



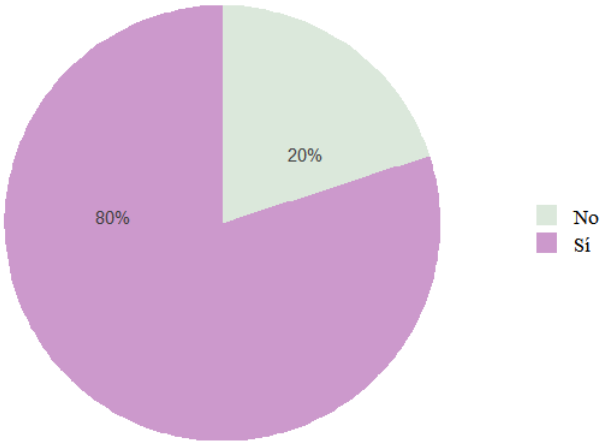
FloorLift

	F. absoluta	%
0	14232	86.4
1	716	4.3
2	544	3.3
3	490	3.0
4	363	2.2
5	124	0.8
6	11	0.1
TOTAL	16480	100.0



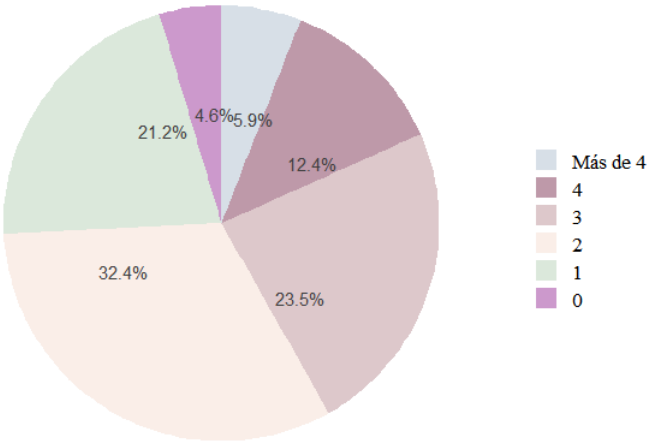
Exterior

	F. absoluta	%
No	3291	20
Sí	13189	80
TOTAL	16480	100



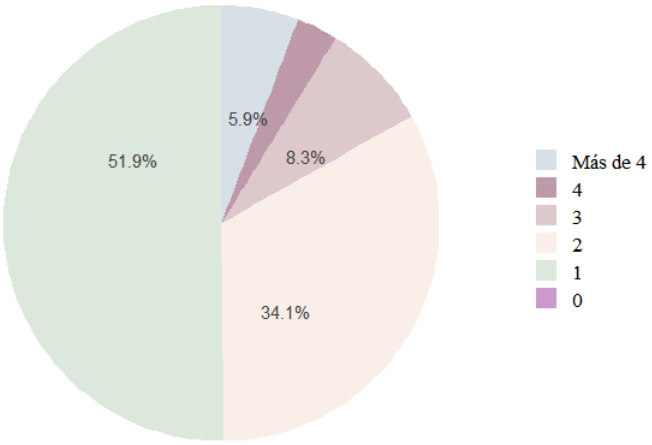
Rooms

Nº Habitaciones	F. absoluta	%
0	751	4.6
1	3492	21.2
2	5346	32.4
3	3878	23.5
4	2050	12.4
5	697	4.2
6	171	1.0
7	63	0.4
8	16	0.1
9	5	0.0
10	4	0.0
11	4	0.0
13	1	0.0
18	1	0.0
20	1	0.0
TOTAL	16480	100.0



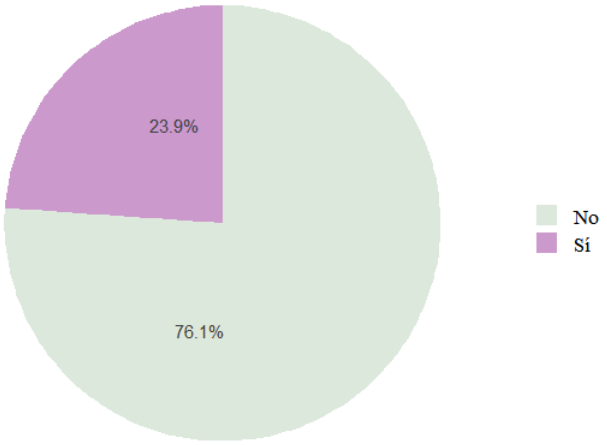
Bathrooms

	F. absoluta	%
0	1	0.0
1	8548	51.9
2	5620	34.1
3	1370	8.3
4	526	3.2
5	282	1.7
6	78	0.5
7	30	0.2
8	13	0.1
9	7	0.0
10	2	0.0
11	1	0.0
12	1	0.0
13	1	0.0
TOTAL	16480	100.0



HasParkingSpace

	F. absoluta	%
No	12534	76.1
Sí	3946	23.9
TOTAL	16480	100.0



ParkingInPrice

	F. absoluta	%
No	13174	79.9
Sí	3306	20.1
TOTAL	16480	100.0

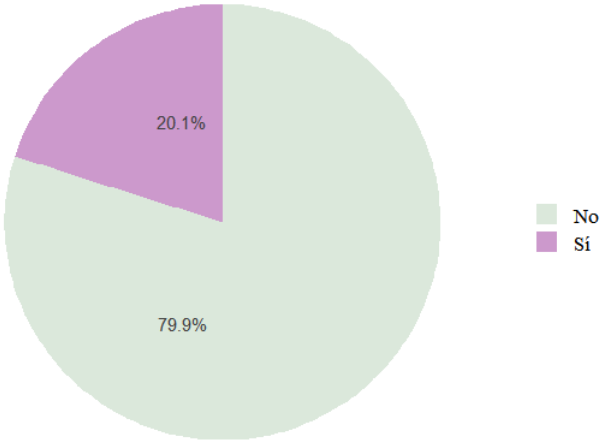
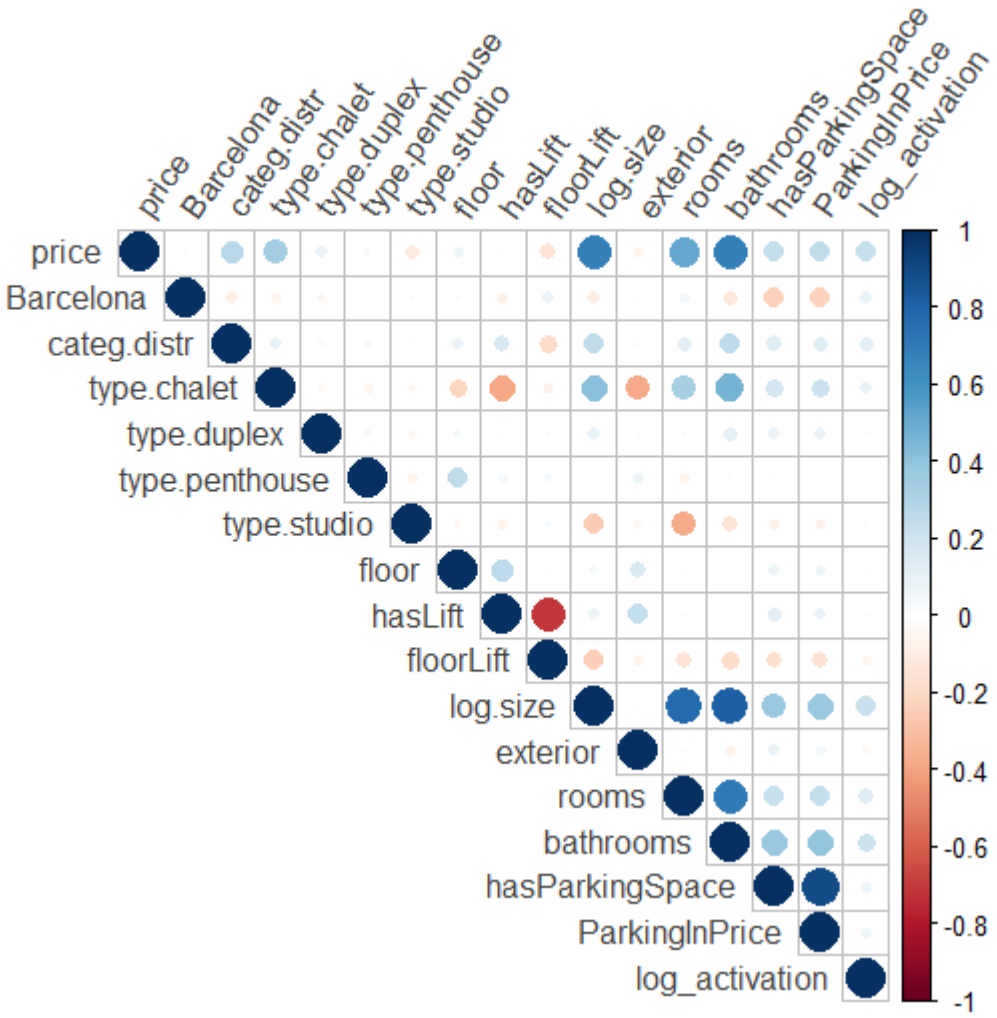


Gráfico de correlaciones



Código R

```
## Idealista

library(nnet)
library(ggplot2)
library(ggthemes)

load(file="rhBM_Price.Rdata")

names(rhBM.price)
log.size <- TRUE
if (log.size){
rhBM.price$size <- log(rhBM.price$size)
names(rhBM.price)[11]<-"log.size"
}

scaled.rhBM.price.tr <- as.data.frame(scale(as.matrix(rhBM.price[Itr,])))
scaled.rhBM.price.tr$price <- rhBM.price$price[Itr]
scaled.rhBM.price.te <- as.data.frame(scale(as.matrix(rhBM.price[Ite,])))
scaled.rhBM.price.te$price <- rhBM.price$price[Ite]


require(caret)
ctrl <- trainControl(
# method = "repeatedcv", # k-fold CV, k=num, repeated repeats times
# repeats = 3, # number of repetitions of the k-fold CV
method = "cv", # k-fold CV, k=num
num=10 # default
)
nnetGrid = expand.grid(size = c(10,15,20), decay = c(0,.1,.3,.5))
set.seed(123)
nnetFit <- train(
log(price)~.,
data=scaled.rhBM.price.tr,
method = "nnet",
tuneGrid = nnetGrid,
trControl = ctrl,
metric = "RMSE",
linout=TRUE
)
nnetFit
nnet.logprice <- nnetFit$finalModel

mod <- Predictor$new(nnet.logprice, data = scaled.rhBM.price.te)

## ALE Plots

eff <- FeatureEffect$new(mod,
feature = "exterior", method = "ale",
grid.size = 30)
```

```
50
51
52 ale_p<-function(var){
53
54 eff$set.feature(var)
55
56 plot(eff) +
57 ggtitle(var) +
58 theme_tufte()
59
60 eff$results
61
62 }
63
64 var<-"Barcelona"
65 ale_p(var)
66
67
68 var<-"categ.distr"
69 ale_p(var)
70
71
72 var<-"type.chalet"
73 ale_p(var)
74
75
76
77 var<-"type.duplex"
78 ale_p(var)
79
80
81 var<-"type.penthouse"
82 ale_p(var)
83
84 var<-"type.studio"
85 ale_p(var)
86
87
88 var<-"floor"
89 ale_p(var)
90
91 var<-"hasLift"
92 ale_p(var)
93
94
95 var<-"floorLift"
96 ale_p(var)
97
98 var<-"log.size"
99 ale_p(var)
100
```

```
101
102 var<-"exterior"
103 ale_p(var)
104
105
106 var<-"rooms"
107 ale_p(var)
108
109 var<-"bathrooms"
110 ale_p(var)
111
112 var<-"hasParkingSpace"
113 ale_p(var)
114
115 var<-"ParkingInPrice"
116 ale_p(var)
117
118
119 var<-"log_activation"
120 ale_p(var)
121
122
123 ## PDP Plots
124
125
126 eff <- FeatureEffect$new(mod,
127 feature = "Barcelona", method = "pdp",
128 grid.size = 30)
129
130
131
132 pdp_p<-function(var, eti){
133
134 eff$set.feature(var)
135
136 plot(eff) +
137 ggtitle(var) +
138 theme_tufte()
139
140 eff$results
141 }
142
143
144
145 var<-"Barcelona"
146 pdp_p(var)
147
148
149 var<-"categ.distr"
150 pdp_p(var)
151
```

```
152
153 var<-"type.chalet"
154 pdp_p(var)
155
156
157
158 var<-"type.duplex"
159 pdp_p(var)
160
161
162 var<-"type.penthouse"
163 pdp_p(var)
164
165 var<-"type.studio"
166 pdp_p(var)
167
168
169 var<-"floor"
170 pdp_p(var)
171
172 var<-"hasLift"
173 pdp_p(var)
174
175
176 var<-"floorLift"
177 pdp_p(var)
178
179 var<-"log.size"
180 pdp_p(var)
181
182
183 var<-"exterior"
184 pdp_p(var)
185
186
187 var<-"rooms"
188 pdp_p(var)
189
190 var<-"bathrooms"
191 pdp_p(var)
192
193 var<-"hasParkingSpace"
194 pdp_p(var)
195
196 var<-"ParkingInPrice"
197 pdp_p(var)
198
199
200 var<-"log_activation"
201 pdp_p(var)
202
```

```
203
204
205 ##  Interación de variables
206
207 interact <- Interaction$new(mod)
208 plot(interact)+theme_tufte()
209
210 interact <- Interaction$new(mod, feature = "Barcelona")
211 plot(interact)+theme_tufte()
212
213 interact <- Interaction$new(mod, feature = "categ.distr")
214 plot(interact)+theme_tufte()
215
216 interact <- Interaction$new(mod, feature = "type.chalet")
217 plot(interact)+theme_tufte()
218
219 interact <- Interaction$new(mod, feature = "type.duplex")
220 plot(interact)+theme_tufte()
221
222 interact <- Interaction$new(mod, feature = "type.penthouse")
223 plot(interact)+theme_tufte()
224
225 interact <- Interaction$new(mod, feature = "type.studio")
226 plot(interact)+theme_tufte()
227
228 interact <- Interaction$new(mod, feature = "floor")
229 plot(interact)+theme_tufte()
230
231 interact <- Interaction$new(mod, feature = "hasLift")
232 plot(interact)+theme_tufte()
233
234 interact <- Interaction$new(mod, feature = "floorLift")
235 plot(interact)+theme_tufte()
236
237 interact <- Interaction$new(mod, feature = "log.size")
238 plot(interact)+theme_tufte()
239
240 interact <- Interaction$new(mod, feature = "exterior")
241 plot(interact)+theme_tufte()
242
243 interact <- Interaction$new(mod, feature = "rooms")
244 plot(interact)+theme_tufte()
245
246 interact <- Interaction$new(mod, feature = "bathrooms")
247 plot(interact)+theme_tufte()
248
249 interact <- Interaction$new(mod, feature = "hasParkingSpace")
250 plot(interact)+theme_tufte()
251
252
253 interact <- Interaction$new(mod, feature = "ParkingInPrice")
```

```
plot(interact)+theme_tufte()

interact <- Interaction$new(mod, feature = "log_activation")
plot(interact)+theme_tufte()


## Permutaciones

library(vip)
vip(nnet.logprice, fill="#800080", alpha=.4, width=.4)+theme_tufte()


## Ghost

function(relev.ghost.out, n1, resid.var, vars=NULL, sum.lm.tr=NULL, alpha
   =.01, ncols.plot=3){
A <- relev.ghost.out$A
V <- relev.ghost.out$V
eig.V <- relev.ghost.out$eig.V
GhostX <- relev.ghost.out$GhostX
relev.ghost <- relev.ghost.out$relev.ghost

p  <- dim(A)[2]

if (ncols.plot<3){
ncols.plot<-3
warning("The number of plot columns must be at least 3")
}
max.plots <- 4*ncols.plot
if (is.null(vars)){
vars <- 1:min(max.plots,p)
}else{
if (length(vars)>max.plots){
vars <- vars[1,max.plots]
warning(
paste("Only the first", max.plots, "selected variables in 'vars' are used")
   )
}
}
n.vars <- length(vars)
nrows.plot <- 1 + n.vars%/%ncols.plot + (n.vars%%ncols.plot>0)

if (!is.null(sum.lm.tr)){
F.transformed <- resid.var*sum.lm.tr$coefficients[-1,3]^2/n1
}
F.critic.transformed <- resid.var*qf(1-alpha,1,n1-p-1)/n1

rel.Gh <- data.frame(relev.ghost=relev.ghost)
rel.Gh$var.names <- colnames(A)

plot.rel.Gh <- ggplot(rel.Gh) +
```

```
geom_bar(aes(x=reorder(var.names,X=length(var.names):1), y=relev.ghost),
stat="identity", fill="darkgray") +
ggtitle("Relev. by ghost variables") +
geom_hline(aes(yintercept = F.critic.transformed),color="blue",size=1.5,
    linetype=2)+
theme(axis.title=element_blank())+
theme_bw()+
ylab("Relevance")+
xlab("Variable name") +
coord_flip()

}


relev.ghost.out <- relev.ghost.var(model=nnet.logprice,
newdata = scaled.rhBM.price.te,
func.model.ghost.var= lm)
```