PLS DISCRIMINANTE: Predictor de la Caída de Precios en Alquileres

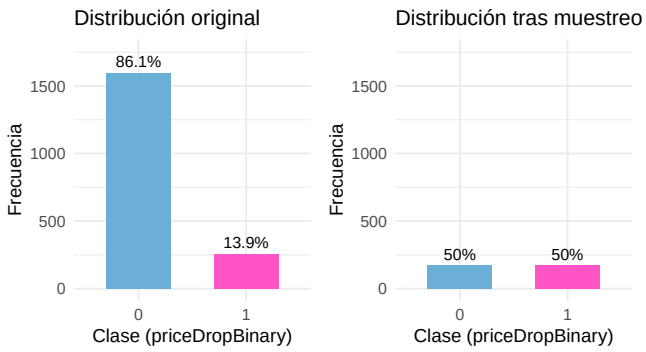
Ana Valiente

2025-06-02

En el análisis anterior, hemos probado que nuestra base de datos no cumple las condiciones de homocedasticidad ni normalidad. Por lo tanto, no podemos aplicar un Análisis Discriminante de Fisher para tratar de predecir qué inmuebles serán rebajados. En su lugar, emplearemos un PLS discriminante; en el que consideraremos como POSITIVOS las viviendas cuyo precio sea menor al precio original. Para ello, convertiremos la variable primitiva priceAmountDrop en priceDropBinary(0 si el precio baja, 1 si se mantiene estable). Excluiremos las siguientes variables del análisis: -URL, lat, lng, zipCode, creationDate (no aportan información relevante) -municipality, neighborhood (incluiremos el cluster servicios en su lugar) -princeAmountDrop (muy relacionada con priceBinaryDrop) Cabe recalcar que la bajada de precio de las viviendas está muy condicionada por el factor humano y las condiciones individuales de cada propietario. Por lo tanto, podríamos considerarla una variable de naturaleza impredecible o, al menos, con un comportamiento difícil de modelar. Aunque no podamos diseñar un predictor infalible, si conseguimos un modelo que prediga las rebajas mejor que el azar, este sería una herramienta útil a la hora de tomar decisiones estratégicas en el mercado inmobiliario.

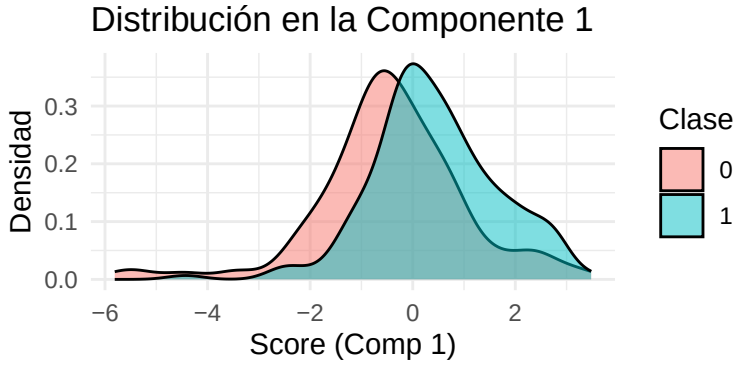
Variables explicativas consideradas en el análisis: [1] “ownerType” “energyEfficiencyRatingType” [3] “environmentImpactRatingType” “bathrooms”
[5] “floor” “hotWater”
[7] “rooms” “surface”
[9] “tieneAscensor” “tieneTrastero”
[11] “tieneCalefaccion” “tieneAireAcondicionado”
[13] “propertySubtypeId” “GeoGeneralRating”
[15] “propertyCounter_buy” “priceDescription_buy”
[17] “propertyCounter_rent” “priceDescription_rent”
[19] “supermarket_count” “pharmacy_count”
[21] “hospital_count” “university_count”
[23] “college_count” “public_transport_count”
[25] “priceAmount” “cluster”

En el gráfico de frecuencias inicial, vemos un claro desbalanceo de las clases (86% N y 14% P). Para evitar obtener un modelo sesgado, inclinado a predecir la clase dominante, efectuaremos un undersampling y entrenaremos el modelo con una base reducida de frecuencias uniformes. Para validar el modelo de forma realista, separaremos entre datos de test y train y, posteriormente, eliminaremos el desbalanceo de la matriz de entrenamiento. En este paso, también escalamos las variables numéricas, puesto que las diferencias en las magnitudes podrían sesgar los resultados del análisis. Una vez separados los datos de prueba y entrenamiento, eliminamos el desbalanceo de estos últimos.



Utilizamos el conjunto de train para generar la primera versión del modelo predictivo. Seguidamente, implementamos la validación cruzada k-folds con k=5 y 10 repeticiones para obtener el número de componentes con el que se minimiza el error de predicción. En nuestro modelo pls-DA, el error es menor con 1 única componente principal. No obstante, destacamos el solapamiento entre los inmuebles que bajan de precio y los que no. Esto nos indica que, seguramente, los grupos no están lo suficientemente diferenciados para realizar predicciones acertadas.

```
## Componentes óptimos: 1
```

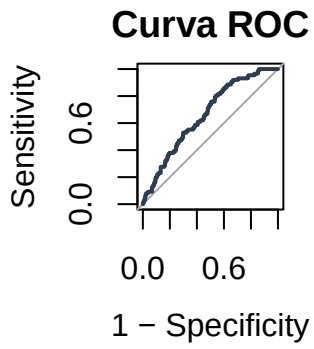


Dado que nuestro objetivo es predecir correctamente si un piso bajará de precio, pero partimos de un conjunto desbalanceado donde la mayoría se mantienen estables, utilizaremos como medidad de calidad la Exactitud Balanceada (balanced accuracy). Esta métrica tiene en cuenta los aciertos en pisos que bajan como en los que no, evitando que el modelo favorezca sistemáticamente la clase mayoritaria. Así, buscamos un rendimiento más justo y representativo que mejore la utilidad práctica del modelo en ambos casos.

```
##          Real
## Predicho   0   1
##        0 300  39
##        1 170  48
```

```
## Accuracy: 0.625
```

```
## Balanced Accuracy: 0.595
```



La curva ROC obtenida para el modelo PLS-DA muestra una capacidad predictiva aceptable, situándose por encima de la línea de referencia (modelo aleatorio). Esto indica que el modelo inicial logra determinar qué pisos bajarán de precio mejor que el azar. Para mejorarlo, calculemos el umbral de clasificación que maximiza la exatitud balanceada:

```
## Umbral óptimo para máxima Balanced Accuracy: 0.446
```

```
##          Real
## Predicho   0   1
##        0 216  17
##        1 254  70
```

```
## Accuracy: 0.513
```

```
## Balanced Accuracy: 0.632
```

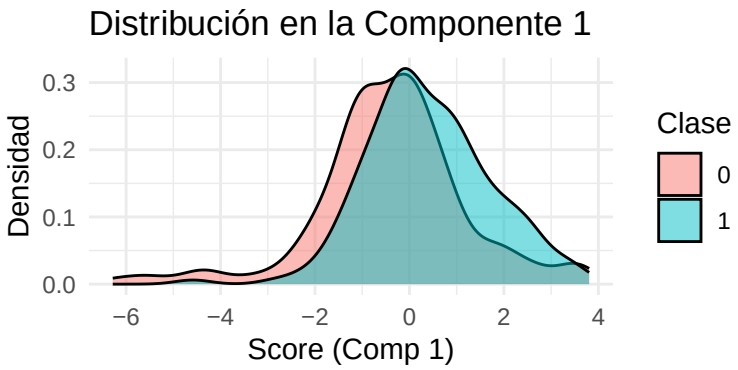
Con el objetivo de potenciar la capacidad predictiva efectuaremos un test de independencia (t para las variables cuantitativas y chi cuadrada o fisher para las cualitativas); para descartar así las variables que no presenten diferencias significativas entre ambas clases. Así, descartaremos la inclusión de variables sin capacidad predictiva, que únicamente generan ruido y disminuyen la fiabilidad de las predicciones.

```
## Variables eliminadas (no presentan diferencias significativas):
```

```
## energyEfficiencyRatingType, environmentImpactRatingType, bathrooms, surface,
## propertyCounter_rent, hospital_count, priceAmount, tieneAscensor,
## tieneTrastero, tieneAireAcondicionado
```

Una vez eliminadas las variables con poca influencia, repetimos el proceso anterior (división y escalado de los datos, generación del modelo y validación) para comprobar si el intento de mejora ha sido efectivo.

```
## Componentes óptimos: 1
```

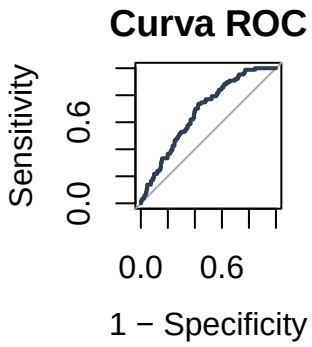


Vemos que todavía obtenemos una distribución relativamente uniforme. Aplicando de nuevo la técnica de validación cruzada con 5 folds y 10 repeticiones, obtenemos 1 componente principal.

```
##          Real
## Predicho   0   1
##        0 305  37
##        1 165  50
```

```
## Accuracy: 0.637
```

```
## Balanced Accuracy: 0.612
```



Vemos que la precisión ha aumentado respecto a su primer valor. Asimismo, en la curva ROC apreciamos que el modelo sigue generando resultados más acertados que los obtenidos por azar. Tal como hemos hecho en el primer modelo, calcularemos el umbral óptimo para tratar de incrementar las medidas de calidad.

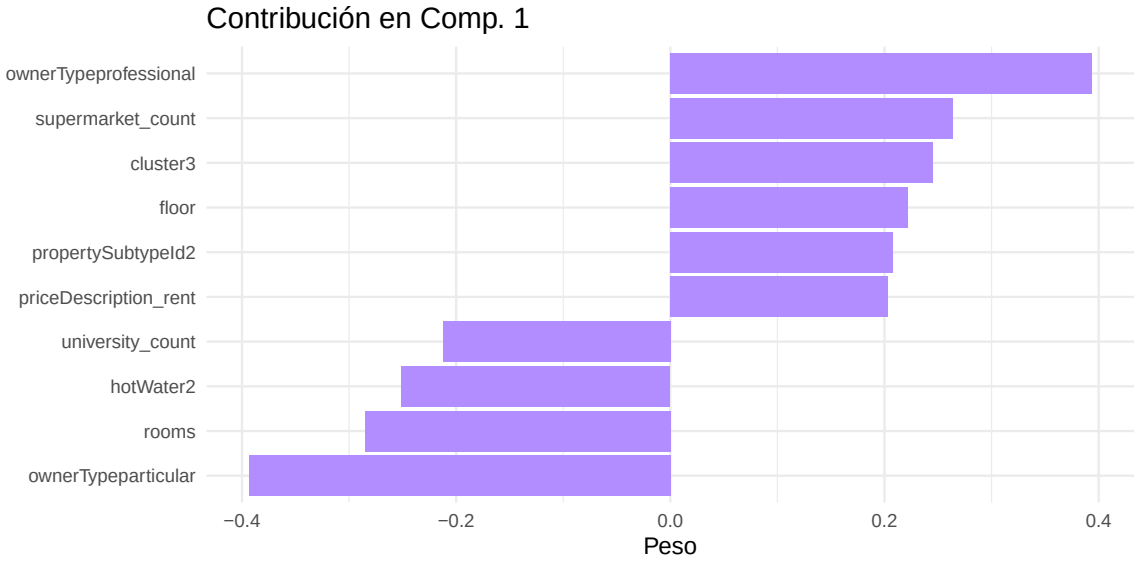
```
## Umbral óptimo para máxima Balanced Accuracy: 0.474
```

```
##          Real
## Predicho   0   1
##        0 270  23
##        1 200  64
```

```
## Accuracy: 0.6
```

```
## Balanced Accuracy: 0.655
```

Finalmente, hemos conseguido un modelo que predice las bajadas de precio de los inmuebles con un 65% de precisión. Esto es, de cada 100 rebajas predichas, 65 serán acertadas. Tal como hemos indicado en el inicio del análisis, esta variable está fuertemente condicionada por factores humanos: decisiones personales de propietarios, estrategias de agencias inmobiliarias, urgencia de venta o negociación, entre otros elementos difíciles de cuantificar. Por tanto, ningún modelo podrá predecirla con exactitud absoluta, y siempre quedarán fuera variables imposibles de medir que condicionarán los resultados. Aun así, cualquier herramienta que logre predecir las fluctuaciones en los precios, aunque sea parcialmente, es altamente valiosa; ya que permite orientar decisiones y ofrecer una ventaja estratégica en el análisis de mercado inmobiliario. Una vez elaborado el modelo definitivo, estudiaremos qué variables tienen más peso en la componente principal para entender qué características hacen a un piso más propenso a ser rebajado.



Resalta claramente que los propietarios profesionales (inmobiliarias, plataformas online, entidades bancarias...) son más propensos a reducir el precio que los propietarios particulares. Por otra parte, si nos centramos en la ubicación, en las zonas de Ruzafa, Benimaclet y Jesús (cluster3) las viviendas son más proclives a las rebajas. Asimismo, cuanto mayor sea el precio medio del alquiler en esta ubicación, más probable será que estos precios bajen. La cantidad de supermercados próximos también contribuye positivamente a dicha probabilidad. Por el contrario, precios de pisos en zonas universitarias son más estables. Respecto a las características del piso per se, las plantas altas tienen más probabilidad de ser rebajadas. En cambio, si tiene muchas habitaciones y/o calentador eléctrico (hotWater2) tiende a experimentar menos variaciones; especialmente en el caso de los apartamentos (porpertySubtype2).