Universidad de Chile
Facultad de Ciencias Físicas y Matemáticas
Departamento Ingeniería Matemática

Laboratorio de Ciencia de Datos

Tarea 1 - EDA y modelos bayesianos

Integrantes: Matías Romero
Danner Schlotterbeck
Kurt Walsen
Profesores: Nicolás Caro
Auxiliares: Rodrigo Lara M.
Fecha: 20 de mayo de 2020
Santiago, Chile

1. Contexto

A continuación se presenta la primera entrega del curso *MA6202 Laboratorio de Ciencia de Datos*. En esta entrega se busca llevar a cabo un problema de ciencia de datos implementando técnicas y metodologías aprendidas en el curso, en particular limpieza de datos, análisis de datos exploratorio(EDA) e implementación de un modelo de regresión lineal bayesiano.

1.1. Objetivo

Se busca establecer una estimación del precio por metro cuadrado para viviendas en la ciudad de Bogotá, Colombia. Además, se busca una estimación de la incertidumbre de las predicciones establecidas.

1.2. Data

Se proporcionan datos recolectados por *web-scrapping* de la página **metrocuadrado**, la cual contiene información sobre venta y arriendo de propiedades, donde se especifican variables como localidad, precio, superficie, número de habitaciones, etc. Además, se proporcionan datos que contienen estadísticas que caracterizan zonas geográficas de interés dentro de la ciudad a estudiar, las unidades que categorizan éstas zonas geográficas llevan por nombre Unidades de Planeación Zonal(UPZ).

2. Carga y limpieza de datos

Se carga la data disponible en el directorio `./data/raw/`, sobre la cual se realizaron una serie de transformaciones con el fin de limpiarla, siguiendo la siguiente metodología:

1. Inicialmente, se concatenan todos los datos por semana estableciendo distinción entre la data proveniente de archivos con 'all' en su nombre con la de archivos con 'furnished' en su nombre. El motivo de ésto es principalmente poder identificar la fuente de cada dato mediante una variable categórica `furnished`, que indica si el dato proviene o no de los archivos 'furnished'. En ambas data notamos la existencia de la variable `url`, correspondiente al enlace de la publicación de venta/arriendo de la propiedad. Dado que existe una clara identificación de un producto con el enlace de su publicación, se decidió utilizar preliminarmente `url` como variable identificadora. Luego, con el uso de la función `DataFrame.query()` se identificó qué datos de archivos 'furnished' no se encontraban en archivos 'all', de lo cual resultaron 4 registros que cumplían tal condición que fueron posteriormente concatenados a la data resultante de la concatenación de los archivos 'all' y nuevamente usando `DataFrame.query()` se generó la columna binaria `furnished`.

2. Se limpia parte de la data, cambiando el formato de ciertas variables de interés:

   - `price` : String de la forma $'\$x'$, con x un número separado por un punto en cada potencia de 10^3. Se decidió expresar la columna como un número flotante, removiendo el signo '$' y los puntos para evitar complicaciones en algunos cálculos.

- `surface` : String de la forma 'x m2', con x un número entero positivo. Se transformaron éstos valores para que representen el valor como un número flotante `float(x)`. Luego, se observó que en esta columna existen valores nulos los cuales corresponden a potenciales inconsistencias y arrastran errores en cálculos futuros, por lo que se decide marcarlos como valores faltantes NaN.
- `n_rooms` : String y float, donde existe un string de la forma '5+'. Se decidió expresar la columna de número de habitaciones como string mediante un replace con un diccionario de transformaciones `x` $\rightarrow$ `str(x)`, pues debido a la forma de los valores en esta columna es posible categorizarlas en base a la cantidad de habitaciones que registra, de esta manera los datos del tipo '5+' serán parte de una única categoría.
- `n_bath` : Análogo a `n_rooms`.
- `property_type|rent_type|location` : String de la forma '{Casa/Apartamento} en {Arriendo/Venta y Arriendo}, {localidad}'. Se decidió dividir esta columna en las 3 variables respectivas mediante una separación por ',' y luego por ' ', donde `property_type` y `rent_type` se reducen a minúsculas y se remueve el prefijo 'en' en esta última. El formato de `location` recibirá un tratamiento más adelante, para generar una variable `barrio`.

3. Se agregan las columnas `price_per_m2` y `n_garajes`, donde la primera representa el precio por metro cuadrado(variable que buscamos regresionar), y la última corresponde al número de garajes que posee la propiedad. La obtención de `price_per_m2` se hizo mediante la relación $\texttt{price\_per\_m2} = \frac{\texttt{price}}{\texttt{surface}}$. La obtención de `n_garajes` requirió buscar en `url` la presencia de la keyword '-garajes', en base a lo cual podemos observar el string que precede a la keyword y así identificar el número de garajes que la propiedad posee. Debido a la existencia de la categoría '4+' dentro de `n_garajes` se decidió establecer las categorías de esta variable como '0','1','2','3','4','4+'. Notamos además la ausencia del keyword '-garajes' en algunas url, por lo que se decidió dejar `n_garajes` como NaN para tales registros.

4. Se clasifican los productos disponibles en la data según el tipo de inmueble al que corresponde y la cantidad de metros cuadrados de superficie que poseen, basándose en la tabla presente en el siguiente **enlace**. Para esto se comenzó definiendo la variable `product_type` y se hizo uso de `query` para la clasificación. Cabe notar que hay entradas en la data que no entran en ninguna de las categorías, dichos índices fueron asignados con el valor `numpy.nan`. Esta clasificación permite fácilmente identificar los datos en un tramo de producto para hacer análisis sobre la variable de respuesta sin tener una relación matemática directa con ésta(caso de `surface`).

5. Se genera una nueva columna `barrio`, a partir de `location`. Se observó que todas las location poseen la estructura '{barrio} Bogotá D.C.', luego se genera `barrio` removiendo 'Bogotá D.C.' del string, además de transformarlo a minúsculas para fijar un formato único.

   Se carga la data disponible en `./data/asignacion_upz/barrio-upz-asignacion.csv`, donde se observó que en la columna `pro_location` existe una mayor variedad de barrios, por lo tanto se utilizó de pivote esta columna para poder hacer el cruce entre la data procesada anteriormente, agregando los códigos UPZ por barrio. Se observó que existen 13 barrios de la data `barrio-upz-asignacion.csv` los cuales no aparecen en la data previamente procesada, ésto se debe quizás a que no hay registros de publicaciones de arriendo/venta de propiedades pertenecientes a tales barrios en nuestra data. Además, se observa que 1946 registros pertenecen a barrios los cuales no se les puede adjuntar un código UPZ. De éstos notamos que la

cantidad única de barrios corresponde a 176 y que un 88 % de las observaciones en la data previamente procesada poseen un código UPZ.

6. Se carga la data disponible en el directorio `./data/estadisticas_upz/` con el fin de enriquecer la data con estadísticas asociadas a los barrios, mediante un cruce de data por medio de código UPZ. Para realizar el cruce es necesario que los códigos UPZ en cada una tengan un mismo formato.

   - `estadisticas_poblacion.csv (upz)` : No presenta diferencia en el formato del código UPZ.
   - `indice_inseguridad.csv (UPlCodigo)` : Presenta en su mayoría formato similar a la data anterior, salvo 4 categorías que fueron indicadas como '1','3','4','5', los cuales se transforman a 'UPZ1','UPZ3','UPZ4','UPZ5' respectivamente.
   - `porcentaje_areas_verdes.csv (cod_upz)` : Presenta un formato numérico de los códigos UPZ, donde se transforma anteponiendo 'UPZ' en todos sus elementos luego de transformarlos a string.

   Luego de cruzar la data, se genera una nueva columna `densidad_poblacion` mediante la relación $\texttt{densidad\_poblacion} = \frac{\texttt{personas}}{\texttt{UPlArea}}$, y luego se borran los últimos posibles duplicados generados luego de los cruces, entre éstos dos de las tres columnas asociadas a los códigos UPZ, donde nos quedamos con `upz` por simplicidad.

3. EDA

Se explora la data previamente limpiada. Previo al análisis fue útil categorizar las variables según el tipo de dato que representan, esto es, variables numéricas, categóricas y misceláneas, mediante una multi-indexación de las columnas. Se decide establecer como variables categóricas a `upz`, `product_type`, `property_type`, `rent_type`, `furnished`, `n_rooms`, `n_bath`, `n_garajes` por su representación en strings junto con no continuidad en el valor de cada una, y como variables misceláneas a `location`, `barrio`, `url`, `details` pues las primeras dos ya se encuentran codificadas mediante `upz` y las últimas dos no corresponden a data que pueda representarse gráficamente o en un modelo. El resto de las variables serán consideradas numéricas.

Considerando como variable respuesta del modelo a `price_per_m2` se realiza el análisis exploratorio, siguiendo la siguiente metodología:

1. Se comienza con un perfilamiento de las variables numéricas, se realiza un histograma para cada variable numérica y se observan ciertos casos de interés, en particular para las variables `price_per_m2` y `surface`(ver Figura (3.1)). En un inicio se observó que sus distribuciones tienen puntos muy alejados de la media, luego se ven como gráficos degenerados donde la mayor cantidad de observaciones se cargan a la izquierda y una cantidad pequeña de valores demasiado alejados de ésta concentración(potenciales outliers) provocan una mala visualización de la distribución. Se observa en la data que los datos que degeneran la distribución de `price_per_m2` corresponden 100 de éstos con $\texttt{price\_per\_m2} >= 10^5$, y para `surface` existe un único dato que degenera la distribución, correspondiente a un dato con `surface`=7400 (el mayor, el dato que le sigue es 540). Removiendo estos 101 datos(0.4 % del total) de nuestra data, notamos una clara mejora en las distribuciones de éstas variables(ver Figura (3.1)).

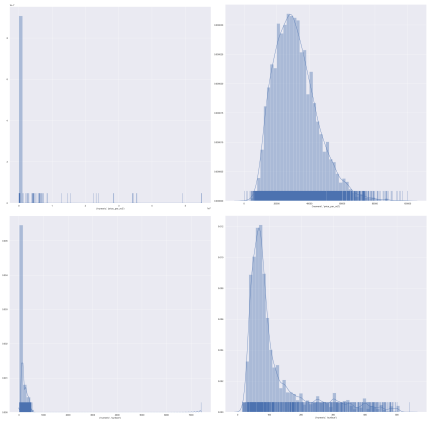

Figura 3.1: Histogramas de `price_per_m2`(arriba) y `surface`(abajo). En la columna izquierda, sin filtrado. En la columna derecha, con filtrado.

Visualizando ahora cómo se comporta la variable `price_per_m2` en respuesta a variables numéricas a través de un regplot, notamos dos casos de interés(ver Figura (3.2)):

- La variable `metrocuadrado_index` presenta una buena distribución en la data y además muestra un comportamiento lineal con algo de ruido, puede que esta variable sea de interés para describir `price_per_m2`.
- Las variables asociadas a estadísticas obtenidas del cruce por `upz` en la sección anterior, por ejemplo `UPlArea`, se muestran como ruido a la hora de describir `price_per_m2`, que no presenta ninguna respuesta clara ante éstas variables. Su significancia deberá ser evaluada más adelante.

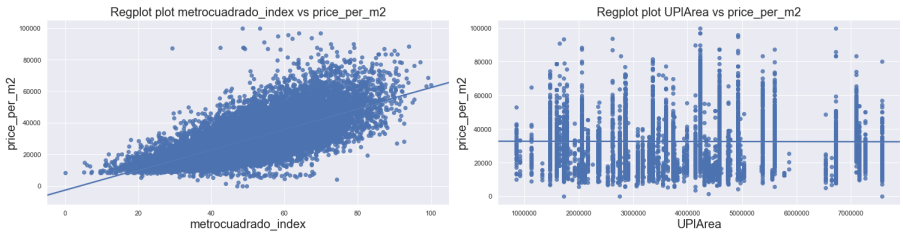

Figura 3.2: Regplots de `metrocuadrado_index`(izquierda) y `UPlArea`(derecha) vs `price_per_m2`

2. Sigue el perfilamiento de las variables categóricas, mediante un histograma por variable se observan ciertos casos de interés(ver Figura (3.3)):

   - Para `product_type` 1-5 (casas), presentan una concentración en valores levemente más bajos que los 6-8 (apartamentos), además la variabilidad del valor de precio dado si es casa o apartamento presentan una distribución similar. Por ende `product_type` podría corresponder a una variable de interés a la hora de definir `price_per_m2`.

- Para `rent_type` notamos que no difieren en mediana y sus distribuciones en la variable `price_per_m2` se comportan de manera similar, por ende no existe una manera de poder identificar una de las categorías en base al valor de `price_per_m2`. Por lo tanto, `rent_type` corresponde a una variable candidata a no ser considerada en el modelo final.
- Para `n_garajes` la diferencia de medias es menos clara, y debido a que los violines son más achatados nos hace dudar sobre la correcta descripción de `price_per_m2` por medio de esta variable. Por medio de un test oneway-ANOVA se corrobora que no hay evidencia estadística que nos permita aceptar la hipótesis de igualdad de medias entre categorías y por lo tanto `n_garajes` corresponde a una variable de interés para el modelo final.

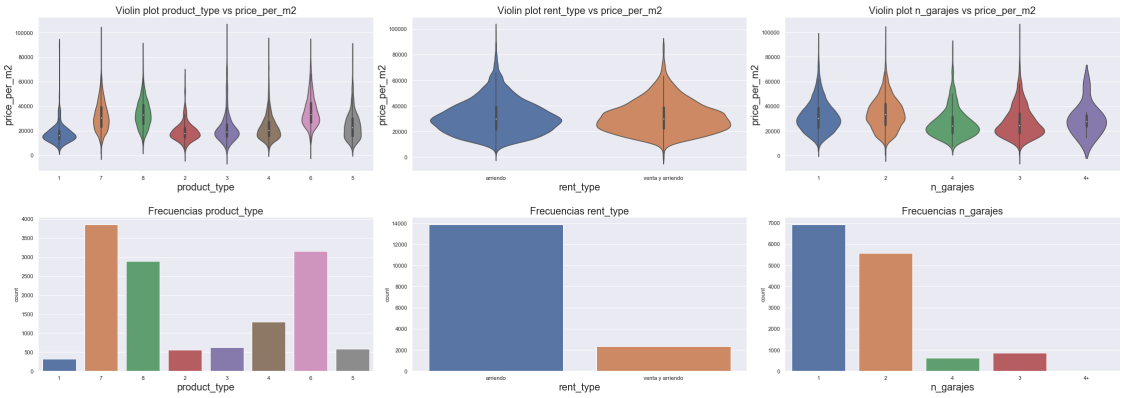

Figura 3.3: Gráficos de violín e histogramas para las variables categóricas `product_type`, `rent_type`, `n_garajes`, respectivamente.

3. Se estudian los datos faltantes. De la Figura (3.4) observamos que las columnas con más datos faltantes corresponden a `product_type` y `n_garajes`.

   - Se puede observar un comportamiento similar en la ausencia de los datos asociados a las estadísticas incluidas mediante un cruce con los códigos UPZ. Ésto claramente es debido a que al existir barrios donde no fue posible obtener el código UPZ, no fue posible cruzar las estadísticas en la sección de limpieza, por lo tanto la ausencia de las estadísticas se refleja en la ausencia de `upz`. Tenemos entonces que ésta variable cumple con la hipótesis MNAR, pues depende de datos no observables en nuestra data. El tratamiento para estos datos faltantes será eliminarlos, pues debido a no poder recuperar `upz`, no podremos recuperar de manera consistente las estadísticas.
   - `product_type` se debe a como la definimos en la sección de limpieza, luego depende de parámetros visibles en la data (`property_type`, `surface`). Luego cumple con la hipótesis MAR, pues depende sólo de datos visibles. El tratamiento para estos datos faltantes será eliminarlos.
   - Para `n_rooms`, `n_bath` no es posible determinar un patrón claro. Se tiene entonces que la información en éstas columnas es perdida completamente al azar, luego ésta perdida de información corresponde al tipo MCAR. Nuestro tratamiento para estos datos faltantes será agrupar por `upz` e imputar por moda.
   - `'n_garajes'` se debe a la ausencia de la keyword '-garajes' en `url`. Por lo tanto, la ausencia de ésta data se puede inferir a partir de la variable `url`, variable sobre la cual

fue construida esta columna. Nuestro tratamiento para estos datos faltantes será agrupar por `upz` e imputar por moda.

- Respecto a las columnas `details`, `price`, `surface` no es posible determinar un patrón claro, más aun cuando los datos faltantes son pocos. Creemos entonces que ésta información es perdida completamente al azar, pues depende de algo que no estamos viendo reflejado en la data(mal ingreso de los datos, omisión de información por parte del vendedor,etc.). Luego cumple con la hipótesis MCAR. Nuestro tratamiento para estos datos faltantes será eliminarlos de la data.

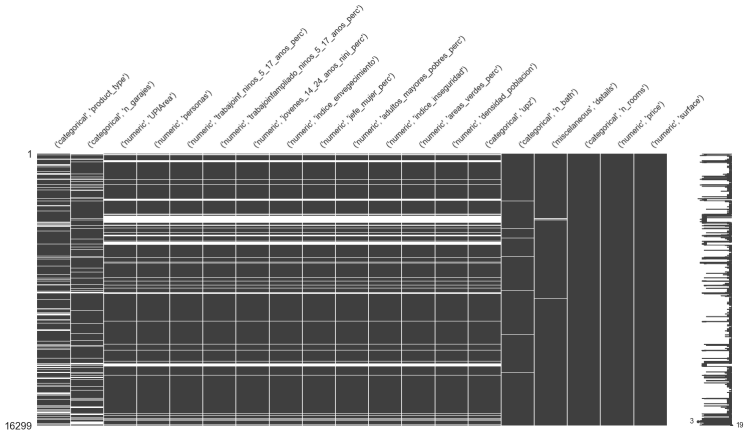

Figura 3.4: Matríz de datos faltantes para la data.

4. Es claro notar que la variable `upz` presenta una alta cantidad de categorías. Por lo tanto, en busca de reducir esta cantidad y obtener una mejor identificación de la variable `price_per_m2` por medio de ésta, se agrupó por `upz` y se obtuvo un valor promedio de `price_per_m2` para cada `upz`, luego se utilizó KMeans para obtener grupos de `upz` en base al precio promedio por metro cuadro, guardando la clasificación de cada `upz` en una nueva variable `upz_cluster`. Se realiza además una clusterizacion con la data filtrada en la sección de perfilamiento univariado numérico. Mediante un test oneway-ANOVA se corrobora que ambos clusters separan estadísticamente bien a la variable `price_per_m2`.

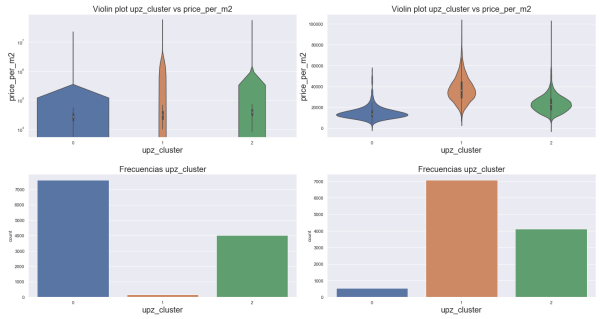

Figura 3.5: Gráficos de violín para `upz_cluster`. A la izquierda, data sin filtrar (escala logaritmica). A la derecha, data filtrada(escala lineal).

Notamos que la clusterización para la data filtrada es considerablemente mejor en cuanto a la identificación de `price_per_m2` mediante ésta, donde cabe destacar que la cantidad de datos que se dropean con tal de mejorar a ésta magnitud las representaciones e identificaciones solo corresponde a un 0.4 % del total. Ésto junto con lo expresado en la sección de perfilamiento univariado es evidencia suficiente para seguir trabajando con la data filtrada, en busca de obtener una buena reprentación de `price_per_m2` sin perder variabilidad en los datos.

5. Se busca cuantificar estadísticamente las relaciones entre variables de interés, comenzando por un análisis de correlación (de Pearson) entre variables numéricas. Se observa en la Figura (3.6) la existencia de grupos de variables con alta correlación (sobre el 70 %), positiva o negativa, por lo que seleccionamos una de cada grupo para evitar colinealidad: descartamos `jovenes_14_24_anos_nini_perc` y `adultos_mayores_pobres_perc` por tener estar estrechamente relacionadas con `indice_envegecimiento`, con correlación de 0.91 y -0.87, respectivamente; y asimismo eliminamos `personas`, por estar correlacionada con `densidad_poblacion` en un 73 %. Corroboramos también que las variables seleccionadas son estadísticamente significativas para la explicación de `price_per_m2`, a través del test de Kolmogorov-Smirnov para verificar que no vengan de una misma distribución, y T-test para la diferencia de medias.

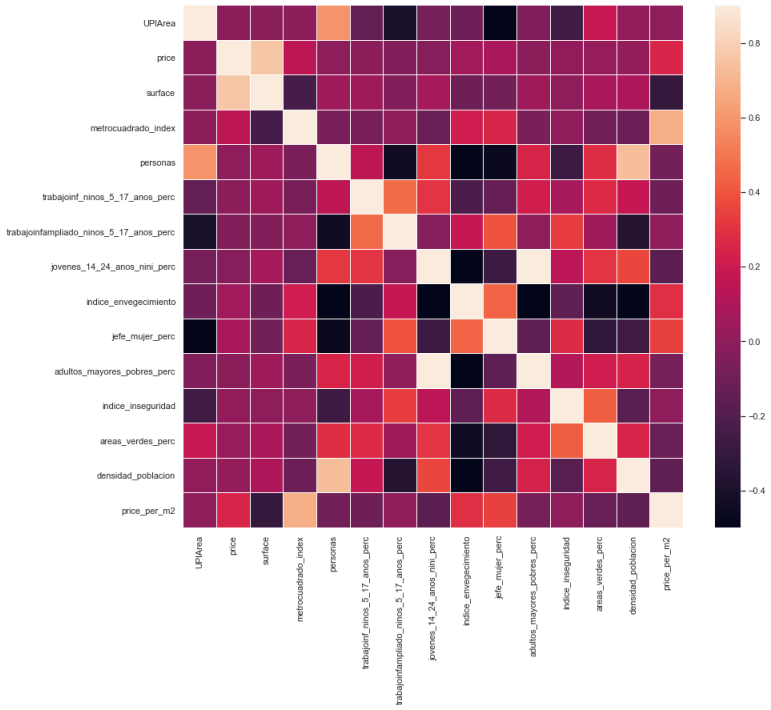

Figura 3.6: Mapa de calor de la matriz de correlaciones.

Para las variables categóricas hacemos una selección de interés: `upz_cluster`, `product_type`, `n_rooms`, `n_bath`, `n_garajes` y `furnished`, donde se determina, mediante one way ANOVA con significancia del 5 %, que todas generan una diferencia significativa entre sus grupos contra la variable objetivo.

6. Se busca detectar observaciones anómalas en la base de datos, para lo cual se utiliza el algoritmo de clustering DBScan sobre el dataset transformado adecuadamente: para las variables

numéricas se utiliza un escalador Min-Máx, ya que ninguna cumple la hipótesis de normalidad, para las categóricas ordinales el transformador OrdinalEncoder, y por último para las no ordinales el transformador OneHotEncoder. Finalmente se aplica DBScan con radio de vecindad $\varepsilon = 0.8$ y cantidad mínima de puntos en la vecindad $N_{min} = 2$, obteniendo cerca de un 7 % de outliers. Observando la Tabla 3.1 se puede apreciar una distribución poco uniforme, lo cual es corroborado por medio de un test χ^2 con 1 % de significancia que rechaza la hipótesis de que las distribuciones sean independientes. De la misma forma, en la Tabla 3.2 se nota una clara concentración de outliers en las casas por sobre los apartamentos, verificado por el mismo test, lo cual se puede deber a que en la gran mayoría de los casos los apartamentos se construyen por grupos, mientras que una casa puede ser única en su tipo, teniendo mayor libertad en sus características.

Étiqueta del clúster UPZ	**0**	**1**	**2**
Porcentaje del total de outliers	52 %	12 %	36 %
Proporción outliers en el cluster	0.11	0.19	0.04

Tabla 3.1: Distribución de outliers por clúster UPZ.

Tipo de producto	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Porcentaje del total de outliers	5 %	12 %	18 %	28 %	18 %	3 %	8 %	8 %
Proporcion de outliers en este producto	0.27	0.33	0.46	0.30	0.46	0.01	0.02	0.03

Tabla 3.2: Distribución de outliers por tipo de producto.

7. Finalmente, dado lo analizado anteriormente se seleccionan las siguientes variables para la realizar la estimación de `price_per_m2`:

   - **Numéricas (8):** `densidad_poblacion`, `metrocuadrado_index`,`jefe_mujer_perc`, `indice_envegecimiento`, `areas_verdes_perc`, `trabajoinf_ninos_5_17_anos_perc`, `trabajoinfampliado_ninos_5_17_anos_perc` y `indice_inseguridad`
   - **Categóricas ordinales (3):** `n_rooms`, `n_bath` y `n_garajes`
   - **Categóricas no ordinales (3):** `product_type`, `upz_cluster` y `furnished`

4. Regresión lineal bayesiana

1. Se comienza creando la clase `RegresionBayesianaEmpirica` la cual implementará el esquema presentado en la parte teórica. Esta clase, además de su `__init__`, consta de los siguientes métodos:

   - `get_posteriori(self, X, y, alpha, beta)`: Este método calcula los objetos m_N y S_N presentados en el esquema iterativo para interactuar con los demás métodos.
   - `fit(self, X, y)`: Este método hace uso del anterior para actualizar los valores de α y β mediante el esquema iterativo presentado. Termina cuando el nivel entregado de tolerancia fue alcanzado por ambos parámetros o al alcanzar el número máximo de iteraciones permitido e imprime un mensaje en la pantalla indicando los valores de los hiperparámetros.

- `predict(self, X_, return_std=False)`: Este método retorna la asignación la media a posteriori a la matriz de observaciones entregada, además si `return_std=True` entrega la desviación estándar asociada a cada una de las observaciones predichas.

2. A continuación se idea un esquema de transformaciones para las variables de forma que el modelo pueda trabajar con valores estandarizados, el flujo consiste en las siguientes transformaciones:

   - Se comienza creando un `Pipeline` dedicado a las variables categóricas no ordinales, el cual las transforma a 'dummies' mediante un `OneHotEncoder`.
   - A continuación el `Pipeline` para las variables ordinales consiste en un `SimpleImputer` que asigna la moda a los valores faltantes en cada variable, seguido de un `OrdinalEncoder` para su codificación. Cabe decir que tras la limpieza de datos antes de entrar al flujo, los únicos valores faltantes se encuentran en variables ordinales, además al codificador le fue entregada la lista de categorías en cada variable ordinal, para asegurar que se mantengan las relaciones de orden.
   - Para las variables numéricas el `Pipeline` consta de un `MinMaxScaler` seguido de `PolynomialFeatures` con grado máximo 3. Es importante destacar que antes de generar el flujo de transformaciones, fue realizado un test de normalidad sobre todas las varaibles numéricas donde la hipótesis de normalidad fue rechazada en todos los casos, es por esto que no se hizo uso del objeto `StandardScaler`.
   - Finalmente se ensamblan los tres flujos anteriores mediante `ColumnTransformer` entegándole a cada transformador la lista de variables sobre las que debe actuar.

   Una vez establecido el flujo de transformaciones, se crea un nuevo `Pipeline` que extiende lo anterior agregando la clase `RegresionBayesianaEmpirica` al final. Adicionalmente se crean dos flujos nuevos: uno restringiendo el conjunto de datos a nuestras variables de interés (para esto basta cambiar la lista de variables sobre las que opera cada transformador y dropear del DataFrame las columnas que no son de interés) y otro cambiando el regresor del final por el objeto `BayesianRidge` del módulo `sklearn.linear_model`.

4.1. Resultados

Los resultados para estos tres flujos en tiempo de ejecución, número de iteraciones utilizadas y R^2 se encuentran en la siguiente tabla. Es importante mecionar que todos los modelos fueron entrenados con los mismos valores iniciales de α, β (10^{-5}) y la misma tolerancia (10^{-9}) además de la misma separación en entrenamiento y testeo para asegurar la comparabilidad.

Modelo	R^2	Tiempo de Ejecución (ms)	N° Iteraciones	α	β
Empírica	0.647	322	2	6.86e-11	1.83e-8
Empírica + sel. variables	0.647	274	2	6.86e-11	1.83e-8
Ridge + sel. variables	0.643	848	300	1.79e-8	3.06e-8

Tabla 4.1: Estadístico R^2, tiempo de ejecución (fitting y score) y número de iteraciones utilizadas para los modelos correspondientes a las partes 3, 4 y 5 del problema.

Se puede observar que en los 3 casos el valor del estadístico R^2 es prácticamente el mismo, alcanzando valores al rededor de 0.64, estos valores son cercanos a los esperados por enunciado y en general considerados en un rango aceptable para un modelo de regresión. Además el hecho que este valor se mantenga al sacar las variables de menor interés refleja que no hay pérdida en la varianza explicada al considerar sólo las variables seleccionadas. Más aún, el tiempo de ejecución disminuyó a un 85 % del inicial, principalmente debido a la reducción de dimensionalidad. Por otro lado, usando `BayesianRidge` el tiempo de ejecución aumentó drásticamente, esto se debe a que usando este estimador a la hora de ajustar el modelo se alcanza el número máximo de iteraciones puesto que la tolerancia en los parámetros no logra ser alcanzada, sin embargo como era de esperar de una librería optimizada, el tiempo promedio por iteración fue mucho menor que usando `RegresionBayesianaEmpirica`. Es importante destacar también que en los dos primeros casos los parámetros α y β dieron exactamente el mismo valor a diferencia del tercero, en el cual se alcanzaron órdenes de magnitud similares sin llegar a converger.

5. Conclusiones

El sector inmobiliario se caracteriza por su alto flujo de oferta y demanda debido a la necesidad básica de vivienda, donde en la elección de ésta participan aspectos clave como la superficie del inmueble, cantidad de baños y habitaciones, sector de residencia y precio. Tener un modelo predictivo sobre el precio por metro cuadrado dado, entre otros, estos aspectos clave resulta de gran utilidad a la hora de, por ejemplo, elegir el rango de precios para futuras propiedades a ser construidas por una inmobiliaria en un determinado sector, mediante un estudio estadístico de la competencia.

Para este proyecto se logró implementar herramientas dispuestas del curso, obteniendo una regresión lineal para describir el precio por metro cuadrado de viviendas de la ciudad de Bogotá mediante la implementación de un modelo de regresión lineal bayesiana empírica. Dicho modelo requirió un manejo de técnicas de análisis de datos como lo son el perfilamiento de variables, análisis de data faltante y métodos de imputación, métodos de agrupación de datos, análisis de observaciones anómalas y formas de tratamiento, entre otros, donde el respaldo de test estadísticos juega un papel importante en dar fundamento a las hipótesis y planteamientos que se generan en el camino. El manejo de `Pipeline` resultó clave para el control del flujo de transformaciones e imputaciones de la data procesada, y además la implementación de la clase `RegresionBayesianaEmpirica` requirió un conocimiento más abstracto sobre el funcionamiento de los métodos `.fit` y `.predict` del modulo `sklearn`, y de un manejo teórico de la forma en la cual estimamos los parámetros.

En el camino se presentaron complicaciones a la hora de realizar el análisis exploratorio, debido a la naturaleza de los datos resultó complicado identificar claramente cuales correspondían efectivamente a anuncios válidos para el análisis, como por ejemplo anuncios de superficies menores a 10 m^2 (prácticamente imposible para una propiedad) o anuncios de arriendo de habitaciones. A ésto se puede agregar que en algunas de las columnas a ser procesadas/limpiadas era reconocible una estructura sobre la cual se podría trabajar en la mayoría de las entradas, sin embargo habían casos particulares en los que dicha estructura no se cumplía por los que había que modificar el esquema de procesamiento previamente concebido. A pesar de esto, la mayoría de estos casos fueron detectados y tratados adecuadamente.

Como futuros pasos se espera ser capaces de ir refinando las técnicas implementadas en esta entrega junto con adquirir nuevas competencias de ciencia de datos, con el fin de generar cada vez análisis más robustos, y así obtener resultados cada vez más significativos.

6. Anexo

6.1. Desarrollo teórico

1. En primer lugar fijemos $\alpha, \beta > 0$ para ahorrar la notación condicional en esos valores. Como los $\{\epsilon_i\}_{i=1}^N$ son normales i.i.d, podemos definir el vector aleatorio $\boldsymbol{\epsilon} = (\epsilon_1, \ldots, \epsilon_N)^T \sim \mathcal{N}(\mathbf{0}, \beta^{-1}\mathbf{I})$, con lo cual el vector de observaciones $\mathbf{y}$ está dado por la fórmula $\mathbf{y} = \mathbf{X}\mathbf{w} + \boldsymbol{\epsilon}$ con verosimilitud

$$p(\mathbf{y}|\mathbf{w}, \mathbf{X}) \sim \mathcal{N}(\mathbf{X}\mathbf{w}, \beta^{-1}\mathbf{I}),$$

donde consideramos la matriz $\mathbf{X} \in \mathbb{R}^{(d-1)\times N}$ que tiene como filas a los vectores $\{\mathbf{x}_i\}_{i=1}^N$. Con esto, podemos reescribir la fórmula de Bayes salvo constante de proporcionalidad:

$$p(\mathbf{w}|\mathbf{y}, \mathbf{X}) \propto p(\mathbf{y}|\mathbf{w}, \mathbf{X})p(\mathbf{w}), \tag{6.1}$$

donde podemos expresar la densidad de $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \alpha^{-1}\mathbf{I})$ (omitiendo las constantes multiplicativas) por

$$p(\mathbf{w}) \propto \exp\left(-\frac{1}{2}\mathbf{w}^T(\alpha^{-1}\mathbf{I})^{-1}\mathbf{w}\right) = \exp\left(-\frac{\alpha}{2}\mathbf{w}^T\mathbf{w}\right),$$

y de manera similar con la verosimilitud de $\mathbf{y}$

$$\begin{aligned} p(\mathbf{y}|\mathbf{w}, \mathbf{X}) &\propto \exp\left(-\frac{\beta}{2}(\mathbf{y} - \mathbf{X}\mathbf{w})^T(\mathbf{y} - \mathbf{X}\mathbf{w})\right) \\ &= \exp\left(-\frac{\beta}{2}\left(\mathbf{y}^T\mathbf{y} - (\mathbf{X}\mathbf{w})^T\mathbf{y} - \mathbf{y}^T(\mathbf{X}\mathbf{w}) + (\mathbf{X}\mathbf{w})^T(\mathbf{X}\mathbf{w})\right)\right) \\ &= \exp\left(-\frac{\beta}{2}\left(\mathbf{y}^T\mathbf{y} - 2(\mathbf{X}\mathbf{w})^T\mathbf{y} + \mathbf{w}^T\mathbf{X}^T\mathbf{X}\mathbf{w}\right)\right) \\ &= \exp\left(-\frac{\beta}{2}\mathbf{y}^T\mathbf{y} + \beta\mathbf{w}^T\mathbf{X}^T\mathbf{y} - \frac{\beta}{2}\mathbf{w}^T\mathbf{X}^T\mathbf{X}\mathbf{w}\right), \end{aligned}$$

donde ocupamos que como $r := (\mathbf{X}\mathbf{w})^T\mathbf{y} \in \mathbb{R}$, entonces $r^T = r$. Juntando estas expresiones en (6.1) obtenemos que

$$\begin{aligned} p(\mathbf{w}|\mathbf{y}, \mathbf{X}) &\propto \exp\left(-\frac{\beta}{2}\mathbf{y}^T\mathbf{y} + \beta\mathbf{w}^T\mathbf{X}^T\mathbf{y} - \frac{\beta}{2}\mathbf{w}^T\mathbf{X}^T\mathbf{X}\mathbf{w} - \frac{\alpha}{2}\mathbf{w}^T\mathbf{w}\right) \\ &= \exp\left(-\frac{\beta}{2}\mathbf{y}^T\mathbf{y}\right)\exp\left(\beta\mathbf{w}^T\mathbf{X}^T\mathbf{y} - \frac{1}{2}\mathbf{w}^T(\beta\mathbf{X}^T\mathbf{X})\mathbf{w} - \frac{1}{2}\mathbf{w}^T(\alpha\mathbf{I})\mathbf{w}\right) \\ &\propto \exp\left(\mathbf{w}^T\beta\mathbf{X}^T\mathbf{y} - \frac{1}{2}\mathbf{w}^T(\beta\mathbf{X}^T\mathbf{X} + \alpha\mathbf{I})\mathbf{w}\right), \end{aligned}$$

donde notamos que el primer término exponencial en la segunda línea no depende de $\mathbf{w}$, por lo que es constante para la densidad a posteriori que estamos calculando. Denotemos $\mathbf{b} := \beta\mathbf{X}^T\mathbf{y} \in \mathbb{R}^{d-1}$ y $\mathbf{S}_N^{-1} := \beta\mathbf{X}^T\mathbf{X} + \alpha\mathbf{I} \in \mathbb{R}^{(d-1)\times(d-1)}$ simétrica e invertible, entonces podemos completar la expresión anterior para que tenga forma de densidad normal multivariada

agregando un término independiente de $\mathbf{w}$:

$$\begin{aligned} p(\mathbf{w}|\mathbf{y},\mathbf{X}) &\propto \exp\left(-\frac{1}{2}\left(\mathbf{w}^T\mathbf{S}_N^{-1}\mathbf{w} - \mathbf{w}^T\mathbf{b} - \mathbf{b}^T\mathbf{w}\right)\right) \\ &\propto \exp\left(-\frac{1}{2}\left(\mathbf{w}^T\mathbf{S}_N^{-1}\mathbf{w} - \mathbf{w}^T\mathbf{b} - \mathbf{b}^T\mathbf{w}\right) - \frac{1}{2}(\mathbf{S}_N\mathbf{b})^T\mathbf{S}_N^{-1}(\mathbf{S}_N\mathbf{b})\right) \\ &= \exp\left(-\frac{1}{2}(\mathbf{w}-\mathbf{S}_N\mathbf{b})^T\mathbf{S}_N^{-1}(\mathbf{w}-\mathbf{S}_N\mathbf{b})\right), \end{aligned}$$

concluyendo así que $p(\mathbf{w}|\mathbf{y},\mathbf{X}) \sim \mathcal{N}(\mathbf{m}_N, \mathbf{S}_N)$, con $\mathbf{m}_N = \mathbf{S}_N\mathbf{b} = \beta\mathbf{S}_N\mathbf{X}^T\mathbf{y}$.

2. Para calcular $p(y'|\mathbf{x}',\mathbf{X})$ notamos que dada una nueva observación $\mathbf{x}'$ podemos deducir que, como $p(\mathbf{w}|\mathbf{y},\mathbf{X}) \sim \mathcal{N}(\mathbf{m}_N, \mathbf{S}_N)$, entonces

$$p(\mathbf{w}^T\mathbf{x}'|\mathbf{x}',\mathbf{y},\mathbf{X}) \sim \mathcal{N}(\mathbf{m}_N^T\mathbf{x}', \mathbf{x}'^T\mathbf{S}_N\mathbf{x}'),$$

al ser una **transformación afín** de un vector normal multivariado. Con esto en mente, podemos aplicar probabilidades totales en la distribución predictiva condicionando sobre $z = \mathbf{w}^T\mathbf{x}$, y usando que $p(y'|z,\mathbf{x}',\mathbf{X}) = p(y'|\mathbf{w},\mathbf{x}',\mathbf{X}) \sim \mathcal{N}(z, \beta^{-1})$ es la verosimilitud de y' obtenemos

$$\begin{aligned} p(y'|\mathbf{x}',\mathbf{X}) &= \int_{\mathbb{R}} p(y'|z,\mathbf{x}',\mathbf{X})p(z|\mathbf{x}',\mathbf{y},\mathbf{X})dz \\ &= \int_{\mathbb{R}} \frac{1}{\sqrt{2\pi\beta^{-1}}}\exp\left(-\frac{1}{2\beta^{-1}}(y'-z)^2\right)\frac{1}{\sqrt{2\pi\mathbf{x}'^T\mathbf{S}_N\mathbf{x}'}}\exp\left(-\frac{1}{2\mathbf{x}'^T\mathbf{S}_N\mathbf{x}'}(z-\mathbf{m}_N^T\mathbf{x}')^2\right)dz. \end{aligned}$$

Considerando las variables aleatorias normales $X \sim \mathcal{N}(0,\beta^{-1})$ y $Z \sim \mathcal{N}(\mathbf{m}_N^T\mathbf{x}', \mathbf{x}'^T\mathbf{S}_N\mathbf{x}')$ podemos reescribir la integral anterior como una convolución de sus densidades, que es justamente la densidad de la suma de estas variables

$$p(y'|\mathbf{x}',\mathbf{X}) = \int_{\mathbb{R}} f_X(y'-z)f_Z(z)dz = f_{X+Z}(y'),$$

y como $X+Z$ distribuye normal con la suma de los parámetros concluimos que

$$p(y'|\mathbf{x}',\mathbf{X}) \sim \mathcal{N}(\mathbf{m}_N^T\mathbf{x}', \beta^{-1} + \mathbf{x}'^T\mathbf{S}_N\mathbf{x}').$$

3. Para este punto vamos a fijar la matriz de observaciones $\mathbf{X}$ para no sobrecargar la notación, y evaluaremos la dependencia sobre α y β. Al aplicar probabilidades totales condicionando sobre $\mathbf{w}$ obtenemos

$$p(\mathbf{y}|\alpha,\beta) = \int_{\mathbb{R}^d} p(\mathbf{y}|\mathbf{w},\beta)p(\mathbf{w}|\alpha)d\mathbf{w},$$

donde $p(\mathbf{y}|\mathbf{w},\beta) \sim \mathcal{N}(\mathbf{X}\mathbf{w}, \beta^{-1}\mathbf{I})$ y $p(\mathbf{w}|\alpha) \sim \mathcal{N}(\mathbf{0}, \alpha^{-1}\mathbf{I})$. Reemplazando estas densidades, la expresión anterior resulta

$$p(\mathbf{y}|\alpha,\beta) = \int_{\mathbb{R}^d} \frac{1}{(2\pi)^{N/2}|\beta^{-1}\mathbf{I}|^{1/2}} \frac{1}{(2\pi)^{d/2}|\alpha^{-1}\mathbf{I}|^{1/2}} \exp\left(-\frac{\beta}{2}(\mathbf{y}-\mathbf{X}\mathbf{w})^T(\mathbf{y}-\mathbf{X}\mathbf{w}) - \frac{\alpha}{2}(\mathbf{w}^T\mathbf{w})\right) d\mathbf{w}$$
$$= \underbrace{\alpha^{d/2}\beta^{N/2}|\mathbf{S}_N|^{1/2}(2\pi)^{-N/2}}_{C} \underbrace{\int_{\mathbb{R}^d} \frac{1}{(2\pi)^{d/2}|\mathbf{S}_N|^{1/2}} \exp\left(-\frac{\beta}{2}(\mathbf{y}-\mathbf{X}\mathbf{w})^T(\mathbf{y}-\mathbf{X}\mathbf{w}) - \frac{\alpha}{2}(\mathbf{w}^T\mathbf{w})\right) d\mathbf{w}}_{I},$$

donde notamos que

$$\log(C) = \frac{d}{2} + \frac{N}{2}\log\beta\log\alpha - \frac{1}{2}\log|\mathbf{S}_N^{-1}| - \frac{N}{2}\log 2\pi,$$

por lo que basta probar que

$$I = \exp\left(E(\mathbf{m}_N)\right) = \exp\left(-\frac{\beta}{2}||\mathbf{y}-\mathbf{X}\mathbf{m}_N||_2^2 - \frac{\alpha}{2}\mathbf{m}_N^T\mathbf{m}_N\right).$$

Notemos que podemos separar la exponencial de la integral:

$$\exp\left(-\frac{\beta}{2}(\mathbf{y}-\mathbf{X}\mathbf{w})^T(\mathbf{y}-\mathbf{X}\mathbf{w}) - \frac{\alpha}{2}(\mathbf{w}^T\mathbf{w})\right) = \exp\left(-\frac{\beta}{2}||\mathbf{y}-\mathbf{X}\mathbf{m}_N||_2^2 - \frac{\alpha}{2}\mathbf{m}_N^T\mathbf{m}_N\right) g(\mathbf{w}),$$

donde el primer término es lo buscado (constante para $\mathbf{w}$) y

$$g(\mathbf{w}) := \exp\left(-\frac{1}{2}(-2\beta(\mathbf{y}-\mathbf{X}\mathbf{m}_N)^T\mathbf{X}(\mathbf{w}-\mathbf{m}_N) + \beta(\mathbf{w}-\mathbf{m}_N)^T\mathbf{X}^T\mathbf{X}(\mathbf{w}-\mathbf{m}_N) + \alpha\mathbf{w}^T\mathbf{w} - \alpha\mathbf{m}_N^T\mathbf{m})\right)$$

puede ser desarrollado para llegar a que

$$g(\mathbf{w}) = \exp\left(-\frac{1}{2}(\mathbf{w}-\mathbf{m}_N)^T\mathbf{S}_N^{-1}(\mathbf{w}-\mathbf{m}_N)\right) = (2\pi)^{d/2}|\mathbf{S}_N|^{1/2} f_W(\mathbf{w}),$$

con $W \sim \mathcal{N}(\mathbf{m}_N, \mathbf{S}_N)$, por lo que

$$I = \exp\left(E(\mathbf{m}_N)\right) \int_{\mathbb{R}^d} f_W(\mathbf{w}) d\mathbf{w} = \exp\left(E(\mathbf{m}_N)\right),$$

concluyendo lo pedido.