Trabajo Práctico N° 1 - Grupo 3

Análisis Exploratorio

El Dataset cuenta con 20 columnas y 460154 registros. A continuación se enumera y detalla cada columna:

- **Id**: Identificador de cada propiedad en el Dataset.
- **Start_date**: Fecha inicio de la operación.
- **End_date**: Fecha final de la operación.
- **Created_on**: Fecha de creación del anuncio en Properati.
- **Latitud**: Coordenada geográfica de la propiedad.
- **Longitud:** Coordenada geográfica de la propiedad.

Observación: Observamos que en los campos donde se menciona la ubicación respecto de un lugar, por ejemplo la provincia, municipio o barrio donde se ubica la propiedad; son cinco campos completados por la persona que creó el anuncio de la propiedad donde no necesariamente completó todos los campos. Generalmente cada campo se completa de la siguiente manera::

- **Place_l2**: Provincia donde se ubica la propiedad.
- **Place_l3**: Departamento/comuna/localidad donde se ubica la propiedad.
- **Place_l4**: Zona o barrio donde se ubica la propiedad.
- **Place_l5**: Zona donde se ubica la propiedad.
- **Place_l6**: Sin información de su contenido pero concluimos que por el nombre también puede referirse a la zona del lugar donde se ubica la propiedad.

Describimos los campos restantes.

- **Operation**: Tipo de operación.
- **Property_type**: Tipo de propiedad. Puede ser casa,departamento, PH, etc.
- **Property_rooms**: Cantidad de habitaciones.
- **Property_bedrooms**: Cantidad de dormitorios.
- **Property_surface_total**: Superficie total en metros cuadrados.
- **Property_surface_covered**: Superficie cubierta total en metros cuadrados.
- **Property_price**: Precio de la propiedad.
- **Property_currency**: Divisa del precio de la propiedad.
- **Property_title**: Título del anuncio de la propiedad en Properati.

Cada variable de las diferentes columnas se pueden dividir en:

- Cuantitativas (numéricas) que pueden ser discretas o continuas.
- Cualitativas (categorías) que pueden ser nominales y ordinales.

En nuestro Dataset se clasifican en:

Observación: En el caso de variables **Start_date**, **End_date** y **Created_on** nos referimos a fechas por lo que de acuerdo al 'supuesto-1' que hicimos, tomaremos las fechas como una variable cualitativa, pues una fecha se puede cuantificar en términos de días, meses y años.

Variables cuantitativas discretas

- Property_rooms
- Property_bedrooms

Variables cuantitativas ordinarias

- Latitud
- Longitud
- Property_surface_total
- Property_surface_covered
- Property_price

Variables cuantitativas discretas

- Id
- Start_Date
- End_Date
- Created_on
- Place_l2
- Place_l3
- Place_l4
- Place_l5
- Operation
- Property_type
- Property_currency
- Property_title

Dado que el objetivo final de este trabajo práctico es el de predecir el precio de una propiedad de acuerdo a las diferentes variables, podemos pensar en primera medida que el precio de una propiedad puede varia de acuerdo a la ubicación, la calidad de la construcción (como esta construida), que tipo de propiedad se vende (una casa, PH, etc) y aspectos que tengan que ver en sí con la propiedad, como la cantidad de habitaciones y el tamaño de dicha propiedad.

Teníamos en mente que tanto la fecha de creación del anuncio (**Created_on**) y la fecha en la que se inició la venta de la propiedad (**Start_date**) pueden influir en el precio puesto que si una propiedad se vende más rápido (la diferencia entre las fechas Created_on y Start_date) que otra, es porque el precio es más barato. Pero para eso primeramente debemos analizar si hay correlación entre las fechas y el precio. Por ahora no lo vemos relevante.

Por otra parte, como el enunciado nos indica que debemos filtrar y solo analizar las propiedades que solo sean de Capital Federal, la variable **Place_l2** (donde generalmente esta) no es relevante.

Pasa lo mismo con la variable **Property_currency** y **Operation** pues solo vamos a filtrar las propiedades que tengan como divisa el dólar y queremos analizar la propiedades que están a la venta.

Otras variables no relevantes son **Property_title** y **Id** dado que no aportan un cambio al precio pues son indicadores para referirse a la propiedad en cuestión.

Como se mencionó anteriormente la variable **Place_l6** es irrelevante pues en su totalidad contiene valores nulos.

Con un análisis posterior vamos a discernir cuál de las variables **Place_l3**, **Place_l4** , **Place_l5** referidas a la ubicación son relevantes.

En conclusión por ahora creemos relevantes las variables Latitud, Longitud, **Property_type, Property_rooms, Property_bedrooms**, **Property_surface_total**, **Property_surface_covered** y **Property_price**.

Preprocesamiento de Datos

1. ¿Se eliminaron columnas (Nombre de la columna y motivo de eliminación?

Al calcular el porcentaje de valores nulos por columna, en los campos "**place_l4**", "**place_l5**" y "**place_l6**" notamos que hay un porcentaje mayor al 50% de valores nulos (**place_l4**: 69.69%, **place_l5**: 99.46% y **place_l6**: 100%), por ende no aportan información así que decidimos directamente eliminar dichos campos del dataset.

En las columnas cuyo tipo de dato sea numérico, los datos nulos fueron reemplazados por su media para imputar dicho dato.

- Al analizar las columqnas property_rooms, property_bedrooms, property_surface_total, property_surface_covered y property_price, vimos que hay anuncios que tienen dormitorios, superficies totales y superficie cubierta sin sentido, por ende tomamos estos datos como datos nulos (datos faltantes).

A los anuncios con el valor de dormitorio sin sentido se les sustituyó el valor con el número de ambientes menos 1 y a los de anuncios con el valor de la superficie total y cubierta sin sentido, por la media.

Por último, notamos que hay d'información que es poco relevante para nuestro problema, como por ejemplo, las columnas 'property_currency' (ya se que todos los anuncios están en dolares), 'operation' (ya se que todos los anuncios están en venta), 'place_l2' (ya se que todos los anuncios están en CABA), 'id', 'start_date', 'end_date', 'created_on' y 'property_title' (Ya que no me aportan una informacion relevante para la investigación),por lo cual decidimos removerlos de nuestro dataset.

2. ¿Detectaron correlaciones interesantes (entre qué variables y qué coeficiente)?

Notamos que entre la variable de dormitorios y ambientes tienden a tener una buena relación lineal, al igual que entre las superficies totales y cubiertas y también entre la cubierta con la cantidad de dormitorios y ambientes. Por ende pueden tener algún tipo de relación entre dichas variables.

3. ¿Generaron nuevos features?

Transformamos los datos no numéricos a datos numéricos:
- En el caso de la columna "property_type", que contiene la zona donde está ubicada el inmueble, aplicamos una transformación One hot encoding. Donde crearemos 2 columnas llamadas type_casa y type_dpto, donde para cada fila (anuncio) se va a cumplir esta restricción:
type_casa + type_dpto <= 1

Si en el campo type_casa del anuncio es igual a 1, eso quiere decir que el anuncio es una propiedad de tipo casa y lo mismo con el type_dpto. En tal caso de que type_casa y type_dpto sean 0 eso quiere decir que la propiedad del anuncio es un PH.

- En el caso de la columna place_l3, la cual contiene el barrio dónde se encuentra la propiedad, observamos que la mayoría se encuentra en los barrios Palermo, Belgrano, Recoleta, Caballito, Villa Urquiza y Almagro y el resto de los inmuebles se encuentran en otras zonas. Por ende decidimos aplicar la transformación one hot encoding a este campo. Donde creamos nuevas columnas, llamadas barrio_Palermo, barrio_Belgrano, barrio_Caballito, barrio_Villa_Urquiza y barrio_Almagro. Ya que dichos barrios conforman mas del 50% de los barrios en que pertenecen los inmuebles. Donde se cumple la siguiente restriccion para cada anuncio:
barrio_Palermo + barrio_Belgrano + barrio_Caballito + barrio_Villa_Urquiza + barrio_Almagro <= 1

Donde el anuncio tiene un valor 1 si se encuentra en los barrios nombrados y si el anuncio no está en dichos barrios, entonces tiene el valor 0 en cada columna.

Se crearon estas columnas ya que creímos importante la información que nos brindaban las columnas *'property_type' y 'place_l3'* y cómo son series de un tipo de dato no número, transformamos la información que nos daba estas columnas a un dato numérico para poder así más adelante tomar en cuenta esta información para el modelo.

4. ¿Encontraron valores atípicos?¿Cuáles?¿Qué técnicas utilizaron y qué decisiones tomaron?

Primero decidimos ver los outliers univariados de cada variable- Para ver si una columna presenta un tipo de dato univariado decidimos ver gráficamente cómo es la distribución de la variable para ver si se puede apreciar una observación con un dato atípico. Y como la mayoría de las variables presentaban un tipo de dato atípico al ver la distribución de la variable quisimos ver gráficamente la relación entre las variables para observar si tienen alguna observación con un outlier multivariado usando mahalanobis.
Observando los gráficos vimos que hay algunas observaciones con algunos valores atípicos que tenían sentido tomarlas en cuenta para el análisis del problema (por ejemplo un anuncio que tiene 10 ambientes y 200m de superficie cuadra) y no nos convenía eliminar dicha observación o imputar algunos datos del registro. Por ende nos fijamos en aquellos outliers y analizamos si podemos tomarlo en cuenta para hacer el estudio, y los que no imputamos aquel valor atípico por su media o lo eliminamos.

5. ¿Qué columnas tenían datos faltantes?¿En qué proporción? ¿Qué se hizo con estos registros?

Las columnas con datos faltantes eran las siguientes columnas:

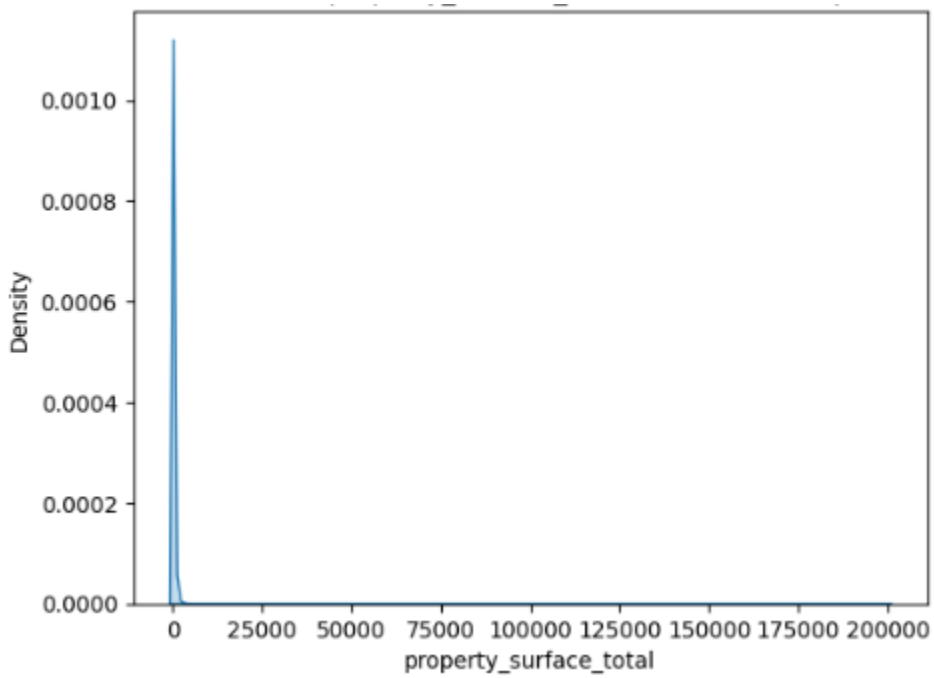
- **latitud** con %8.76 de datos faltantes
- **longitud** con %8.76 de datos faltantes
- **place_l2** con %0.00 de datos faltantes
- **place_l3** con %4.86 de datos faltantes
- **place_l4** con %69.69 de datos faltantes
- **place_l5** con %99.46 de datos faltantes
- **place_l6** con %100.00 de datos faltantes
- **operation** con %0.00 de datos faltantes
- **property_type** con %0.00 de datos faltantes
- **property_rooms** con %19.86 de datos faltantes
- **property_bedrooms** con %25.21 de datos faltantes
- **property_surface_total** con %13.54 de datos faltantes
- **property_surface_covered** con %7.01 de datos faltantes
- **property_currenc**y con %4.04 de datos faltantes

Aquellos registros que contenían una columna numérica con dato faltante, lo que se procedió a imputar dicho campo del registro por su media. Y a los registros que contengan un dato faltante en una columna no numérica, se intentó aplicar una regresión lógica para predecir el valor, pero se descubrió que al intentar esto se alcanzaba solo un 15% de aciertos, consecuentemente se trabajó con la información contenida en otras columnas, como el título de la propiedad, para alcanzar la información buscada con los datos ya obtenidos y para el resto de los santos faltantes se utilizó el porcentaje de distribución para cada barrio para completarlo sin cambiar las tendencias de estos.

Visualizaciones

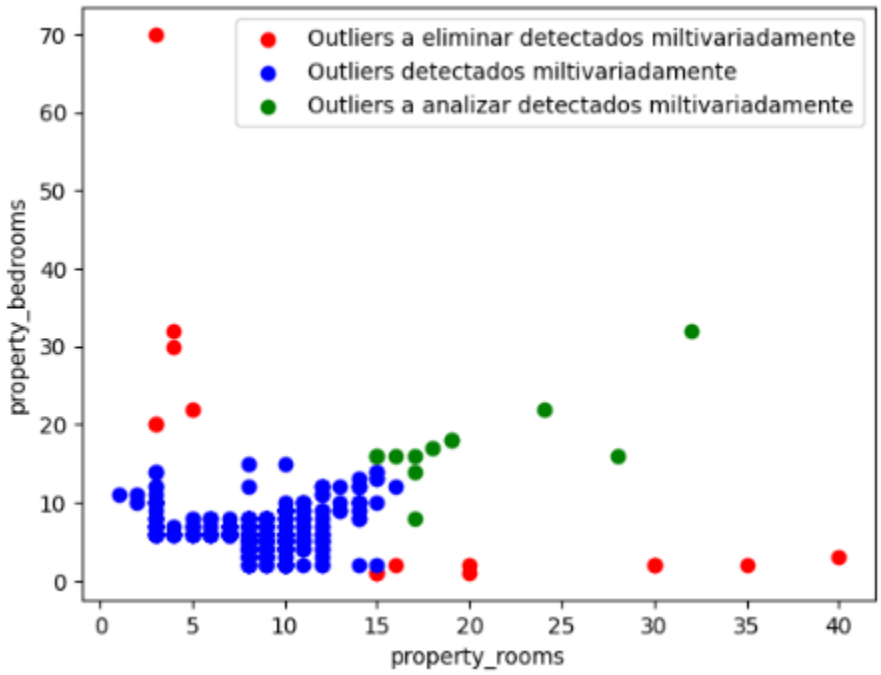
A continuación se mostrarán dos gráficos que elegimos para mostrar la distribución de los datos.

Distribución de la variable de la superficie total antes de detectar outliers



Nos pareció interesante la distribución de esta variable porque se puede detectar a simple vista que hay algunos outliers y nos da a entender rápidamente que hay algo raro en esas observaciones ya que no tiene sentido que una propiedad tenga 200000 metros cuadrados de superficie. Por ende gracias a este tipo de gráficos podemos darnos cuenta que hay algo que analizar en la variable.

Distribución y Outliers de los ambientes y de las habitaciones



Este gráfico también nos pareció muy interesante ya que nos aporta a simple vista varias informaciones. Se puede apreciar todos los outliers multivariados detectados usando mahalanobis y podemos apreciar que hay algunos outliers detectados que no tiene sentido eliminar dichas observaciones porque nos pueden restar algún tipo de información en el análisis que queremos tomar en cuenta. Hay outliers que básicamente harían ruido en mi conjunto datos por ende hay que eliminarlos, y hay otros outliers que podemos analizar dichas observaciones y analizarlos a ver si tienen algún sentido tomarlos en cuenta. Y además en el gráfico podemos apreciar aquella relación entre las variables que tienden a ser una relación lineal.

Agrupamiento

Debido a que hay que analizar la tendencia del clustering se decidió hacer un gráfico de dispersión para analizar la relación que existe entre dos variables del dataset.
Al observar el gráfico se puede encontrar relaciones interesantes entre dos variables. Consideramos las que tienen más tendencia a agrupamiento para analizar.

1. longitud-latitud
2. longitud-place_l3
3. latitud-place_l3
4. type_property - placa_l3
5. property rooms-place_l3
6. property bedroom -place_l3
7. property room -property bedroom
8. property room- property_surce-total
9. type_property - property_surce-total

Para analizar la tendencia de agrupamiento se utilizó el algoritmo de K-Means en las 9 relaciones propuestas anteriormente. Se calculó el coeficiente de silhouette para diferentes valores de K (números de clusters) utilizados en el algoritmo de K-Means.

1. **longitud-latitud**

Una de las relaciones más importantes que consideramos fue la de latitud-longitud donde se podía observar la ubicación de cada propiedad que estaba a la venta en C.A.B.A. Para esta relación se recomienda tomar 8 centroides para el agrupamiento. Por lo que se pueden agrupar 8 distintos lugares en donde se encuentran más propiedades a la venta. Dado que el dataset muestra que los barrios donde se venden más propiedades son en Palermo, Belgrano, Caballito, Recoleta, Villa Urquiza, Almagro y Nuñez, creemos que es una agrupación acertada.

2. **Longitud-place_13**

Esta relación es con cada coordenada geográfica y barrio de C.A.B.A. Se observa que se recomienda utilizar 6 centroides para agrupar. Esto tiene sentido pues al ser la latitud una coordenada geográfica pueden haber dos propiedades a la venta que tengan una ubicación con una misma latitud y diferente latitud.

3. **Latitud-place_13**

Lo mismo pasa con la dependencia del barrio y la longitud pues, pueden haber propiedades que se encuentren ubicadas con una misma latitud pero a diferente longitud, También se nos recomendo 6 como la cantidad de centroides apropiados para el agrupamiento.

4. **type_property - placa_l3**

Expresa la relación que existe entre el tipo de propiedad y el barrio. El K recomendado en este caso fue 6. Respecto a este agrupamiento no encontramos un criterio de agrupación dado que, con los grupos coloreados con el K recomendado, en cada agrupamiento de barrios se venden tanto casas, departamentos y PHs. Existe una relación lineal entre cada tipo de propiedad y los barrios.

Un criterio pensado fue que en cada barrio hay una tendencia en vender un tipo de propiedad. Por ejemplo, en barrios como Recoleta, o Palermo hay una mayor tendencia a vender departamentos, mientras que en barrios como Mataderos hay una mayor cantidad de casas a vender.

5. **property bedroom -place_l3**

Expresa la relación que hay entre la cantidad de habitaciones y barrios. El k recomendado fue 4. Se observa que existe una tendencia de grupos de barrios donde se venden propiedades con más habitaciones que en otros grupos..

6. **property bedroom -place_l3**

Expresa la relación que hay entre la cantidad de dormitorios y barrios. Se observa lo mismo que en la relación property bedroom -place_l3.

7. **property room -property bedroom**

Expresa la relación entre la cantidad de dormitorios y ambientes. El K recomendado es 6. Generalmente la cantidad de dormitorios y ambientes está estrechamente relacionado con la superficie total de la propiedad. Por lo que sí hay mayor cantidad de habitaciones, puede aumentar la probabilidad de encontrarse con más dormitorios. No tienen una tendencia a agruparse.

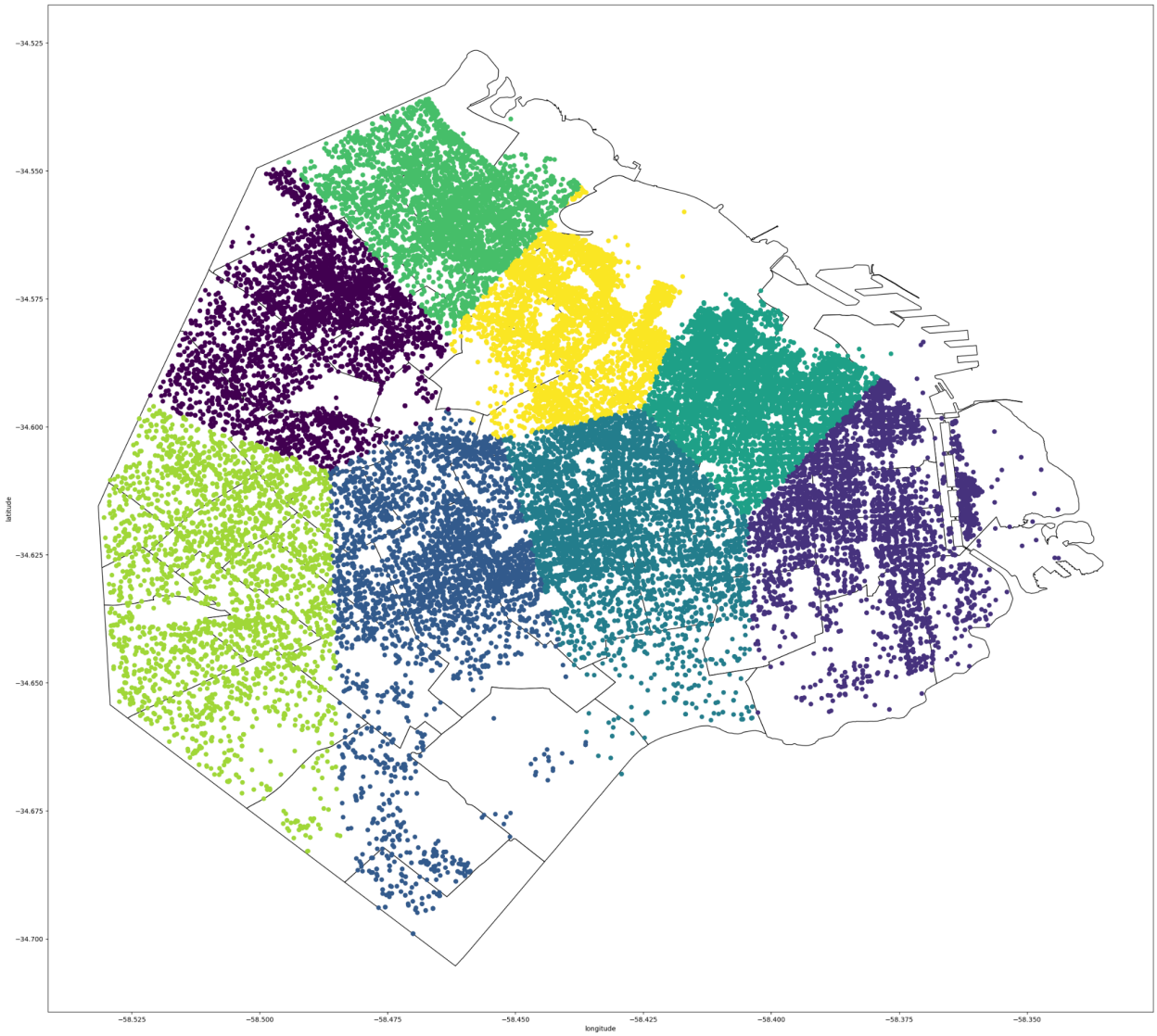
8. property room- property_surce-total

Expresa la relación entre cantidad de habitaciones y la superficie total. El K recomendado es 2. Se observa que existen dos grupos en el cual la cantidad de superficie total se mantiene pero la cantidad de habitaciones aumentan y, el otro grupo, en el cual aumenta la cantidad de superficie total y se mantiene la cantidad de habitaciones. Un criterio para esto puede ser que el primer grupo se deban más a departamentos y ph y el segundo grupo de casas generalmente.

9. type_property - property_surce-total

Expresa la relación entre el tipo de propiedad y la superficie total. El K recomendado fue dos. Se observa que se formaron dos grupos donde el primer grupo ,conformado por departamentos, tiene más superficie total que el segundo grupo (conformado por departamentos, casas y ph).

Mapa de C.A.B.A con 8 grupos recomendados de acuerdo a K-Means


−34.525
−34.550
−34.575
−34.600
−34.625
−34.650
−34.675
−34.700
latitude
−58.525
−58.500
−58.475
−58.450
−58.425
−58.400
−58.375
−58.350
longitude

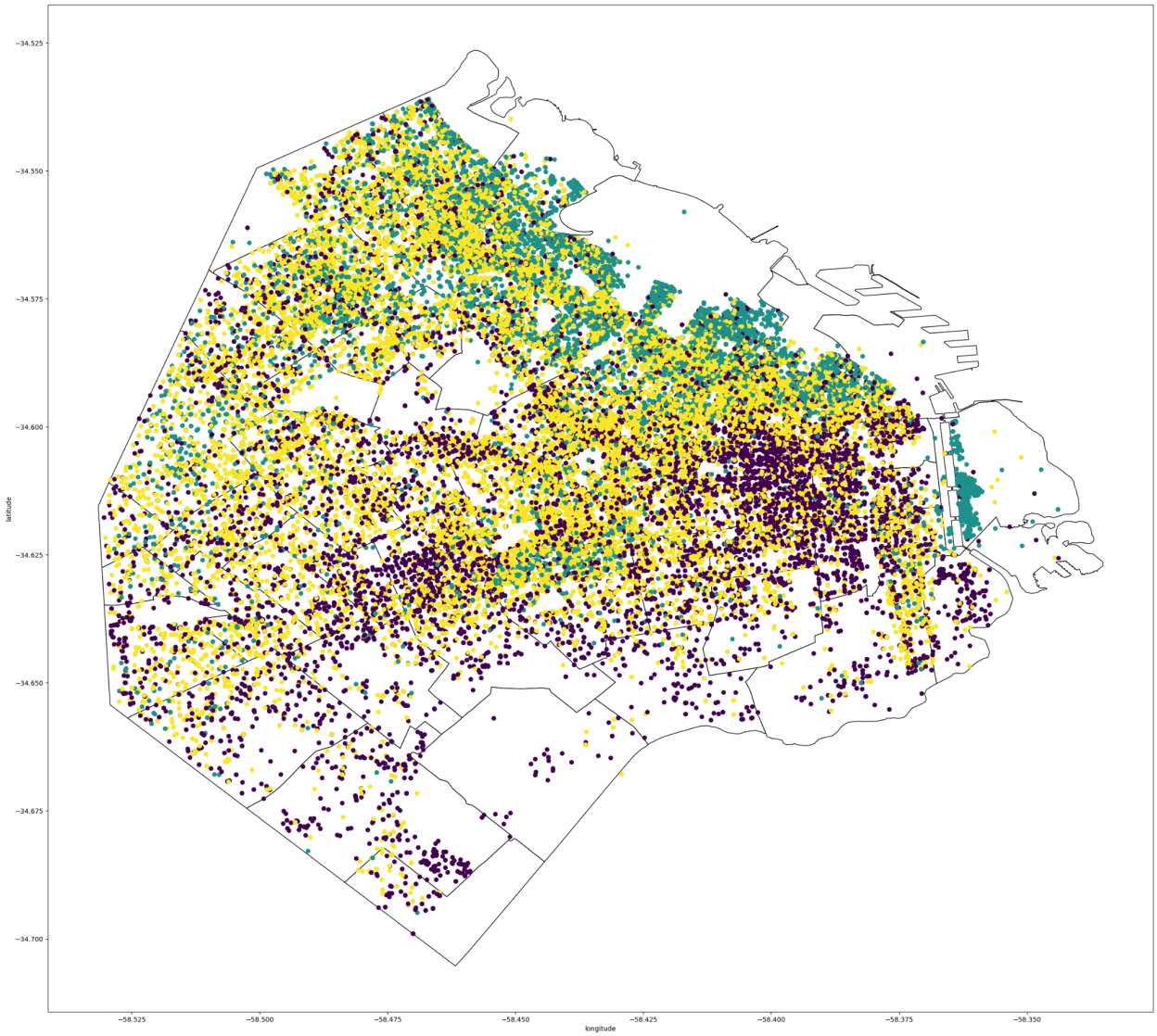
Clasificación

Con el objetivo de clasificar los anuncios de viviendas por su precio se creará una nueva variable **tipo_precio**, la cual tendrá tres categorías: alto, medio y bajo. Para determinar el tipo_precio de una propiedad se analizará el precio por metro cuadrado **pxm2.**

Dado que para el cálculo total del precio de una propiedad se tienen en cuenta diferentes factores incluyendo el tipo de propiedad que se está vendiendo y como el cálculo de **pxm2** se realiza del cociente entre el precio total estipulado por el vendedor y la superficie total, elegimos la opción 3. Otra razón de la elección fue que tienen un coeficiente de correlación igual a 0.25. En nuestro caso y por lo dicho antes, lo consideramos importante.

Luego, viendo el gráfico de dispersión obtenido por el algoritmo K-Means con 3 centroides, se observa que existe una superposición entre los 3 grupos. Esto es un resultado esperado pues para diferentes tipos de propiedad pueden tener un mismo precio por metro cuadrado ya que, por lo mencionado anteriormente, en el precio se consideran diferentes factores incluyendo el tipo de propiedad.

Mapa de C.A.B.A con los avisos coloreados por la variable tipo_precio


−34.525
−34.550
−34.575
−34.600
−34.625
−34.650
−34.675
−34.700
latitude
−58.525
−58.500
−58.475
−58.450
−58.425
−58.400
−58.375
−58.350
longitude

a. Construcción del modelo

Arbol de Decision

El árbol de decisión es un modelo predictivo en donde se dividen un conjunto de datos en subconjuntos más pequeños y homogéneos con respecto a una variable objetivo. Los nodos del árbol representan características o atributos y las aristas representan reglas de decisión basadas en esos atributos.

- ¿Optimizaron hiperparámetros? ¿Cuáles?

Se optimizaron los siguientes hiperparámetros:

1. función para medir la calidad del división (Ginni, Entropía)
2. la profundidad máxima del árbol (max_depth)
3. la poda (ccp_alpha).

- ¿Utilizaron K-fold Cross Validation?¿Cuántos folds utilizaron?

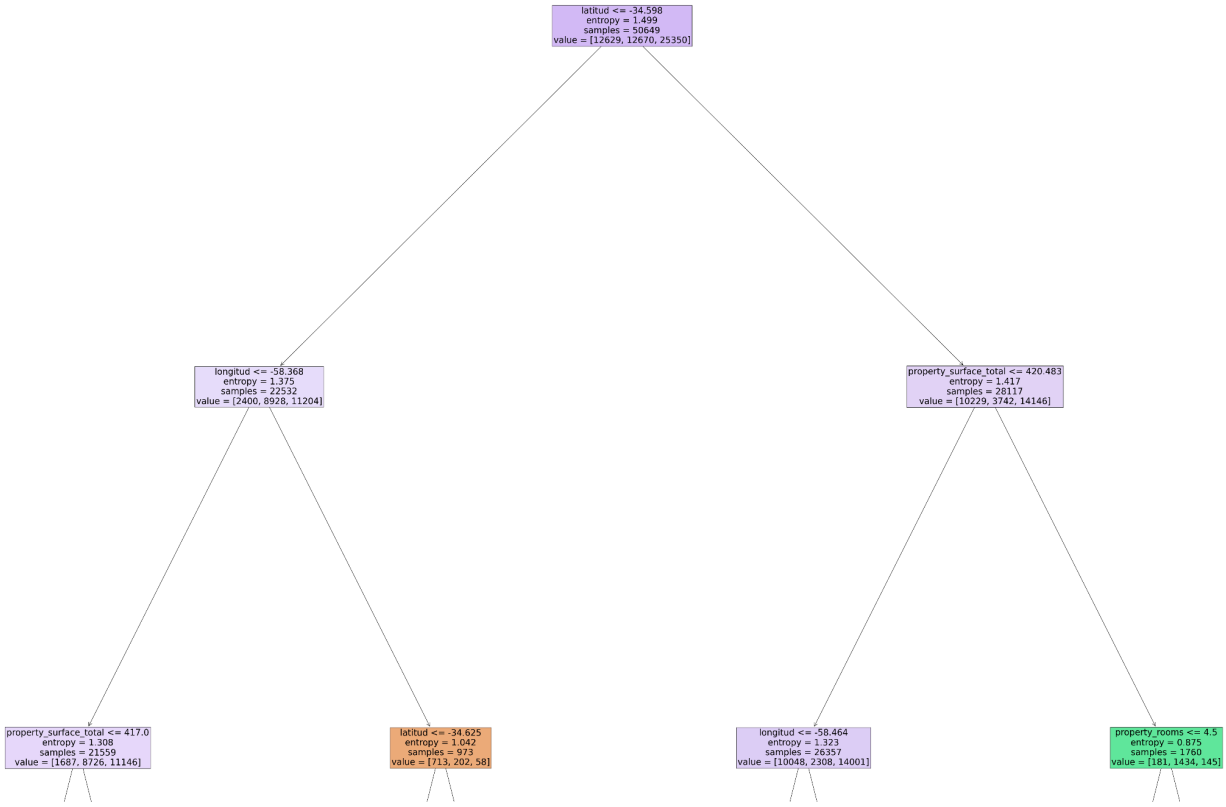
Si, se uso K-fold Cross Validation con 10 folds.

- ¿Qué métrica utilizaron para buscar los hiperparámetros?

La métrica para buscar los hiperparámetros que se utilizo fue f1-score.

- Añadir imagen del árbol generado e incluir descripciones que consideren adecuadas para entender el mismo. Si es muy extenso mostrar una porción representativa.

Porción representativa del árbol de decisión.



Vemos que el árbol de decisión tomó a la latitud como atributo más importante, es decir es el atributo que más aporta información al árbol (la cual tiene la menor entropía que el resto de los atributos por el momento).

Lo primero que hará el árbol es dividir el dataset en dos subconjuntos, para esto se fija primero en el valor de la latitud de cada observación, colocara en un subconjunto las observaciones con la latitud <= a -34,598 y en otro las observaciones > a -34,598.

Luego, los atributos que aportan más información para los dos subconjuntos creados son la latitud y la superficie total (property_suface_total).

Al subconjunto con la latitud <= -34,598 lo dividirá en dos subconjuntos más y para ello se fija en el valor de la longitud, separando las observaciones con una longitud <= a -58,368 en un subconjunto y las observaciones con la longitud > -58,368 en otro subconjunto.

Para el subconjunto con la latitud que tiene > -34.598 lo dividirá en dos subconjuntos fijándose en el valor de la superficie total de cada observación, las observaciones que tienen una superficie <= a 420,481 los junta en un subconjunto y el resto de las observaciones en otro subconjunto.

Random Forest

Es un algoritmo de aprendizaje automático que se basa en la técnica de ensamblaje de modelos , específicamente en el ensamblaje de árboles de decisión, que se construyen en forma paralela. Cuando se realiza una predicción, cada árbol emite una predicción, entre todas las predicciones elegidas, se elige el mejor clasificador.

- ¿Optimizaron hiperparámetros? ¿Cuáles?

Se optimizaron los siguientes hiperparámetros:

1. función para medir la calidad del división (Entropía)
2. el número mínimo de muestras requeridas en un nodo hoja (min_samples_leaf)
3. el número mínimo de muestras requeridas para dividir un nodo (min_samples_split)
4. el número de árboles en el ensamble (n_estimators).

- ¿Utilizaron K-fold Cross Validation?¿Cuántos folds utilizaron?

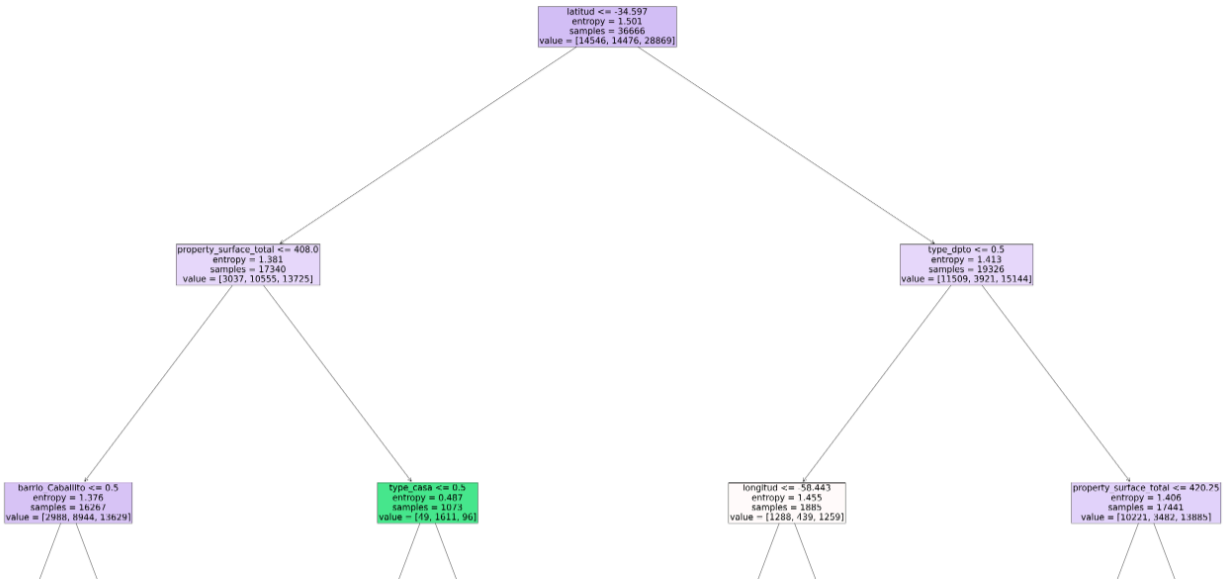
Si, se uso K-fold Cross Validation con 10 folds.

- ¿Qué métrica utilizaron para buscar los hiperparámetros?

La métrica para buscar los hiperparámetros que se utilizo fue f1-score.

- Mostrar la conformación final de uno de los árboles generados. Si es muy extenso mostrar una porción representativa y explicar las primeras reglas.

Porción representativa de uno de los árboles generados.



Vemos que el árbol tomó como el atributo que aporta más información la latitud, donde va a dividir el conjunto de datos aquellas observaciones que tienen una latitud menor o igual a -34,597 los coloca en un subconjunto donde luego se fijará en el atributo que aporta más información (en ese subconjunto) que en ese caso es el atributo de la superficie total donde va a separar las observaciones que tengan una superficie total menor o igual a 408 con las observaciones que tengan una superficie total mayor a 408.

Para el conjunto de datos que tenga una latitud mayor a -34,597 se va a fijar en el atributo que aporta más información para ese conjunto de datos, donde ese atributo es el type dpto, donde me va a separar las observaciones son de tipo departamento y los que no son de tipo departamento.

Regresion Logisca

Es un método estadístico utilizado para modelar y analizar relaciones entre una variable dependiente binaria (categórica con dos categorías) y una o más variables independientes. La regresión logística se aplica cuando se desea predecir si una observación pertenece a una de las dos categorías.

- ¿Optimizaron hiperparámetros? ¿Cuáles?

No se optimizaron hiperparametros.

- ¿Utilizaron K-fold Cross Validation?¿Cuántos folds utilizaron?

Si, se uso K-fold Cross Validation con 10 folds.

- ¿Qué métrica utilizaron para buscar los hiperparámetros?

La métrica para buscar los hiperparámetros que se utilizo fue f1-score.

Cuadro de Resultados

Realizar un cuadro de resultados comparando los modelos que entrenaron (entre ellos debe figurar cuál es el que seleccionaron como mejor predictor).

Medidas de rendimiento en el conjunto de TEST:

- F1
- Precision
- Recall
- Accuracy

Confeccionar el siguiente cuadro con esta información:

Modelo	F1-Test	Precision Test	Recall Test	Accuracy Test
Arbol de Decision	0.687610	0.687788	0.688247	0.688247
Random Forest	0.702186	0.720370	0.708215	0.708215
Regresion Logistica	0.529297	0.597299	0.575624	0.575624

En cada caso ¿Cómo resultó la performance respecto al set de entrenamiento?

Nota: indicar brevemente en qué consiste cada modelo de la tabla

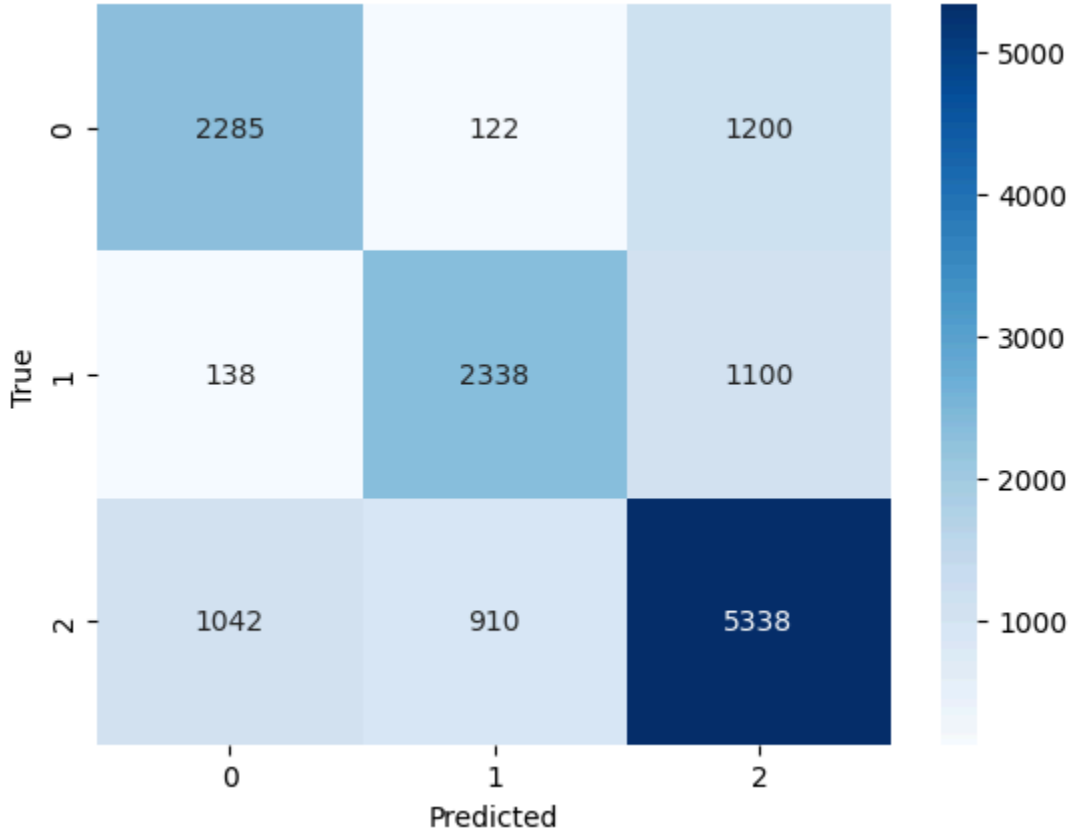
1) Árbol de decisión:

Mejores hiperparámetros: `{'max_depth': 22, 'criterion': 'entropy', 'ccp_alpha': 6.31578947368421e-05}`

Mejor métrica f1-score obtenida en el entrenamiento: `0.687758`
F1-score obtenida en test: `0.687610`

Subió en menor medida el valor de f1-score en la performance respecto del entrenamiento.

Matriz de confusión para el modelo del Árbol de decisión



A partir de la matriz de confusión se observa que:

- Verdaderos positivos: en total se encontraron 9961 muestras tanto de propiedades con precio bajo, medio y alto donde la predicción fue correcta.
- Falsos positivos: en total se encontraron 4512 muestras donde la predicción fue incorrecta al clasificar propiedades en precios bajos, medianos y altos .

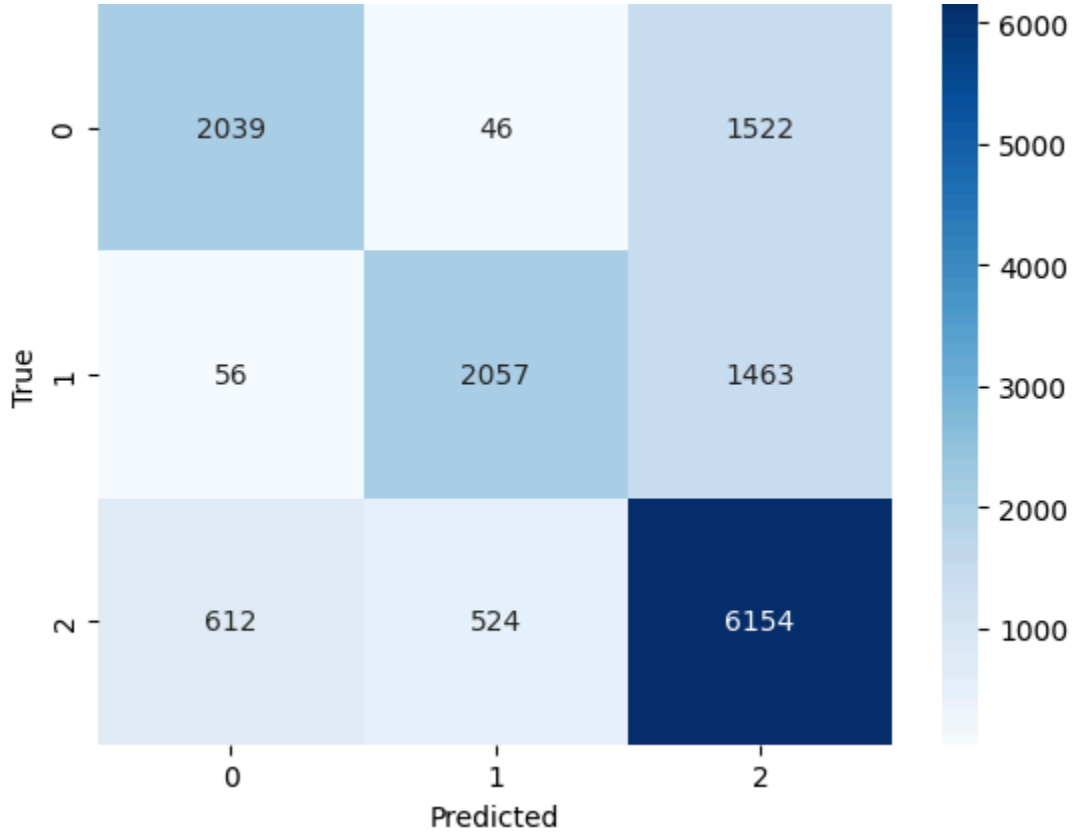
2) **Random Forest:**

Mejores hiperparámetros: `{'n_estimators': 46, 'min_samples_split': 5, 'min_samples_leaf': 7, 'criterion': 'entropy'}`

Mejor métrica f1-score obtenida en el entrenamiento: `0.706984`
F1-score obtenida en test: `0.702186`

Bajo casi nada el valor de f1-score en la performance respecto del entrenamiento.

Matriz de confusión para el modelo de Random Forest



A partir de la matriz de confusión se observa que:

- Verdaderos positivos: en total se encontraron 10250 muestras tanto de propiedades con precio bajo, medio y alto donde la predicción de la clasificación fue correcta.

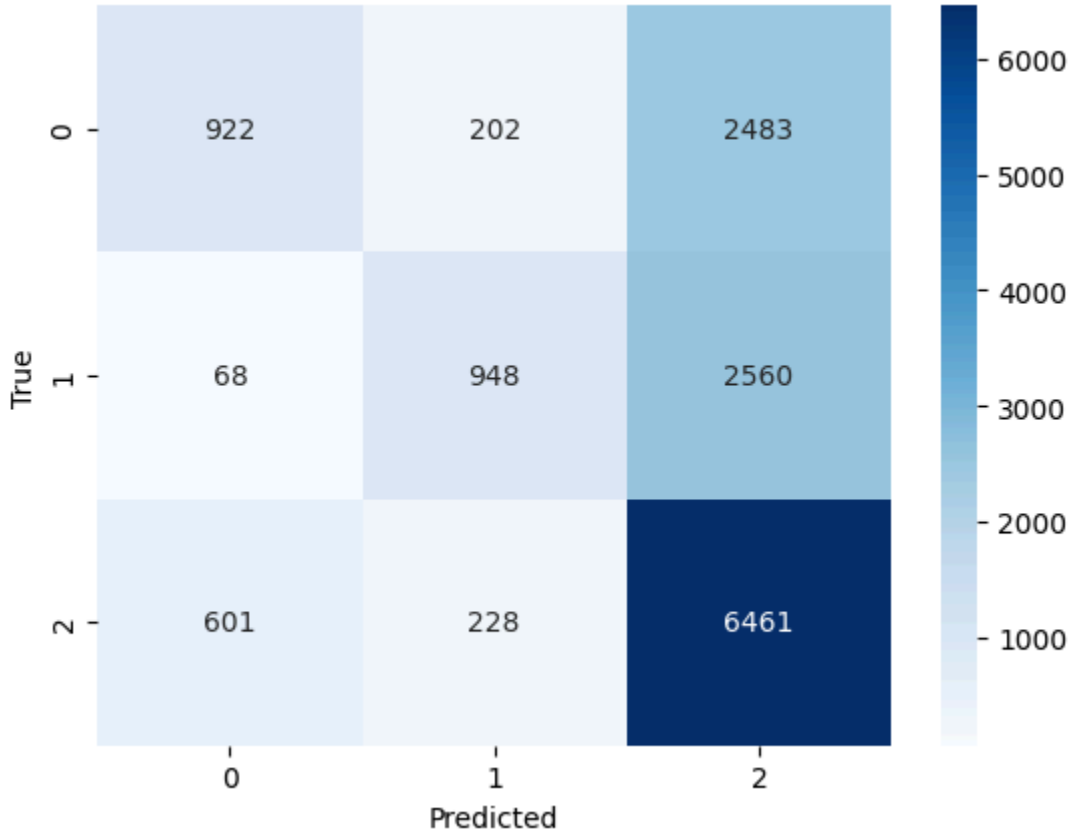
- Falsos positivos: en total se encontraron 4223 muestras donde la predicción fue incorrecta al clasificar propiedades en precios bajos, medianos y altos .

3) Regresión logística:

Hiperparámetros: `(cv=10, max_iter=10000, n_jobs=-1, random_state=1, scoring=make_scorer(f1_score, average=micro), tol=1e-05)`

F1-score obtenida en test: `0.529297`

Matriz de confusión para de Regresión logística



A partir de la matriz de confusión se observa que:

- Verdaderos positivos: en total se encontraron 8331 muestras tanto de propiedades con precio bajo, medio y alto donde la predicción fue correcta.
- Falsos positivos: en total se encontraron 6142 muestras donde la predicción fue incorrecta al clasificar propiedades en precios bajos, medianos y altos .

c. Elección del modelo

Dado los diferentes scores de f1-score en test, era de esperarse que la regresión logística tenga el peor score entre los 3.
Por otra lado, al comparar las métricas en test del modelo del Árbol de decisión y Random Forest, los mejores scores obtenidos son de Random Forest.
Por otra parte, si comparamos la variación del f1-score en entrenamiento y la de test, f1-score varía menos en el modelo de Árbol de decisión.

Dado que Random Forest tiene los mejores scores de todas la métricas por una gran diferencia, en conclusión, el mejor modelo que predice a la hora de clasificar una observación en la categoría de tipo precio (bajo, medio y alto) es Randon Forest.

Metricas

Las métricas utilizadas en el trabajo práctico son:

Precisión: es la relación entre los verdaderos positivos (casos correctos) y el total de predicciones positivas (verdaderos positivos más falsos positivos).

Recall: es la relación entre los verdaderos positivos (casos correctos) y el total de casos verdaderamente positivos y verdaderamente negativos.

Accuracy: es la relación entre los casos predecidos correctamente (verdaderos positivos y verdaderos negativos) y el total de todas las predicciones.

F1-score: es una relación que combina la precisión y recall.

Error Cuadrático Medio (Mean Squared Error): un estimador que mide el promedio de los errores al cuadrado, es decir, la diferencia entre el estimador y lo que se estima**.** Mientras más chico más cerca está nuestro modelo de los datos reales. Al ser un valor elevado al cuadrado, es sensible a valores de diferencias grandes.

Raíz del Error Cuadrático Medio (Root Mean Squared Error): es una medida de uso frecuente de las diferencias entre los valores (valores de muestra) predichos por un modelo o un estimador y los valores observados**.** Mide el desvío estándar (cuánto se alejan los valores de la media).

Error Absoluto Medio (MAE): un estimador que mide el promedio de los errores , es decir, la diferencia entre el estimador y lo que se estima**.**

Regresión

Si llegaron a entrenar alguno de los modelos, mencionar cuáles y qué métricas obtuvieron en test y si realizaron nuevas transformaciones sobre los datos (encoding, normalización, etc) completando los ítems a y b:

a. Construcción del modelo

KNN

Su funcionamiento se basa en la idea de que objetos similares tienden a estar cerca en el espacio de características. Dado un punto, el algoritmo predice o clasifica según la mayoría de puntos cercanos a él.

- ¿Utilizaron K-fold Cross Validation?¿Cuántos folds utilizaron?

Si, se uso K-fold Cross Validation con 10 folds.

- ¿Qué métrica utilizaron para buscar los hiperparámetros?

La métrica para buscar los hiperparámetros que se utilizo fue MAE.

- Evaluar la performance del modelo en el conjunto de evaluación, explicar todas las métricas. Comparar con la performance de entrenamiento.

Mejores hiperparametros obtenidos: `{'weights': 'uniform', 'p': 1, 'n_neighbors': 21, 'metric': 'chebyshev', 'leaf_size': 29, 'algorithm': 'ball_tree'}`

Mejor métrica MAE obtenida en entrenamiento: `70669.18423838983`
MAE obtenida en test: `76742.2672689928`

Subió en gran medida el valor de MAE en la performance respecto al entrenamiento.

XGBoost

Está basado en boosting y construye un conjunto de árboles de forma secuencial, y cada uno se enfoca en corregir los errores de los anteriores. Incluye términos de regularización en la función de pérdida para controlar el sobreajuste y mejorar la generalización del modelo.

- ¿Utilizaron K-fold Cross Validation?¿Cuántos folds utilizaron?

Si, se uso K-fold Cross Validation con 10 folds.

- ¿Qué métrica utilizaron para buscar los hiperparámetros?

La métrica para buscar los hiperparámetros que se utilizo fue MAE.

- Evaluar la performance del modelo en el conjunto de evaluación, explicar todas las métricas. Comparar con la performance de entrenamiento.

Mejores hiperparametros obtenidos: `{'subsample': 0.6206896551724138, 'n_estimators': 150, 'max_depth': 2, 'learning_rate': 0.7747738693467336, 'lambda': 10.0, 'gamma': 1.894736842105263, 'eta': 0.5000000000000001}`

Mejor métrica MAE obtenida en entrenamiento: `55663.54823947504`
R2 obtenida en test: `62092.09001991371`

Subió en gran medida el MAE en la performance respecto del entrenamiento.

Método a elección: Gradient Boost

Está basado en boosting y construye un conjunto de árboles de forma secuencial, y cada uno se enfoca en corregir los errores de los anteriores. Incluye términos de regularización en la función de pérdida para controlar el sobreajuste y mejorar la generalización del modelo.

- ¿Utilizaron K-fold Cross Validation?¿Cuántos folds utilizaron?

Si, se uso K-fold Cross Validation con 10 folds.

- ¿Qué métrica utilizaron para buscar los hiperparámetros?

La métrica para buscar los hiperparámetros que se utilizo fue MAE.

- Evaluar la performance del modelo en el conjunto de evaluación, explicar todas las métricas. Comparar con la performance de entrenamiento.

Mejores hiperparametros obtenidos: `{'max_depth': 1, 'loss': 'huber', 'criterion': 'squared_error'}`

Mejor métrica r2 obtenida en entrenamiento: `68309.0556354723`
R2 obtenida en test: `74425.37505393875`

Subió el valor de MAE en la performance respecto del entrenamiento.

b. Cuadro de Resultados

Realizar un cuadro de resultados comparando los modelos que entrenaron (entre ellos debe figurar cuál es el que seleccionaron como mejor predictor).

Medidas de rendimiento en el conjunto de TEST:

- MSE
- RMSE
- MAE

Confeccionar el siguiente cuadro con esta información:

Modelo	MSE	RMSE	MAE
KNN	46609295330.97465	215891.86027030906	76742.2672689928
XGBoost	28785232539.25582	169662.11285745507	62092.09001991371
Gradient Boost	58601681095.59148	242077.84098424102	74425.37505393875

c. Elección del modelo

Teniendo en cuenta las medidas RMSE, MSE y MAE, el modelo XGBoost es muchísimo menor que la medida dada por los otros modelos, aunque su variación de error absoluto medio , en entrenamiento y test,es un poco mayor que los otros modelos.
Entonces, el modelo que predice mejor es XGBoost.

Conclusiones Finales

A lo largo del trabajo hecho sobre el dataset provisto por Properati observamos varias cosas, entre las principales se encuentran las siguientes:

En las etapas de análisis exploratorio y preprocesamiento de datos notamos que había valores faltantes, en aquellas columnas donde había un alto porcentaje de valores faltantes directamente se eliminaban ya que no aportan información al análisis del problema, y aquellas variables que son de un tipo de dato numérico se imputo por su media, y en la variable de los barrios se utilizó para los datos faltantes un modelo de clasificación. Transformamos la variable del tipo de propiedad utilizando la técnica de one hot encoding. Y para la columna de los barrios también se usó la misma técnica pero antes se vio que barrios tenían mayor porcentaje en el conjunto de datos y solo se colocaron dichos barrios, aquellos anuncios que no están ubicados en dichos barrios lo tomamos como que está ubicado en otro barrio. También notamos algunos outliers univariados en las superficies y en la cantidad de ambientes y dormitorios. Para ello se asociamos las variables ambientes con

dormitorios y superficie total con la superficie cubierta, y se utilizó la técnica de la distancia mahalanobis y se colocó un umbral, se observó aquellas observaciones que superan dicho umbral y se analizó si tendrían sentido, es decir si un inmueble tiene una superficie alta y tiene una cantidad de dormitorios importante (algo que pueda tener sentido) y viceversa, entonces dichas observaciones se consideraron no modificarlas y tenerlas en el dataset, pero aquellas observaciones que no tienen sentido se modificó los campos que tienen valores atípico por su media ya que podrían ocasionar ruido en el dataset. También se analizó las columnas más relevantes para el análisis del problema y aquellas que no eran relevantes (`'property_currency'`, `'operation'`, `'property_title'`, `'place_l2'`, `'id'`, `'start_date'`, `'end_date'`, `'created_on'`) se eliminaron del data set.

En la etapa de agrupamiento fue difícil buscar una forma de agrupar variables que no tengan correlación entre ellas. Para ello nos fijamos en el gráfico de scatter las relaciones entre variables que tengan una mayor distribución y que creemos interesantes. Analizamos nueve relaciones de las cuales, en la mayoría, no encontramos un criterio en conjunto para agruparlas . Un criterio de agrupación interesante es la relación entre la latitud y la longitud que, sumado al gráfico del mapa de C.A.BA., podíamos observar cómo se distribuyen la venta de las propiedades en toda capital. Otro criterio que pensamos fue la relación que existe con los barrios y el tipo de propiedad, en esta relación podríamos agrupar las diferentes áreas en donde se venden más departamentos, PHs y casas en C.A.B.A. Se observa que hay barrios donde hay una tendencia a vender más departamentos, que puede deberse a que son barrios pertenecientes a zonas de C.A.BA. más céntricas como por ejemplo Palermo. Por otra parte, hay barrios que tienen a vender más casas como en las zonas cercanas al conurbano bonaerense.

Por último, en las etapas de clasificación y regresión vimos que es importante optimizar los hiperparámetros de los modelos, ya que estos afectan significativamente la performance de estos y que los modelos que a priori pensábamos que iban a ser los mejores fueron efectivamente los mejores. Por otro lado, en la parte de clasificación vimos que la variable del precio por metro cuadrado depende bastante de no solo el tipo de propiedad, sino también de su ubicación y pensamos que tal vez haber elegido un target relativo a la ubicación además de el tipo de propiedad nos hubiera dado mejores resultados en performance de los modelos.

De las cosas que quedaron fuera del tp, nos hubiera gustado analizar los campos de latitud y longitud de los inmuebles para después predecir en dónde estaría ubicado un inmueble.

Una segunda cuestión que hubiéramos querido explorar era usar los datos de las columnas de **Created_on** (fecha de la creación de anuncio de la propiedad en Properati) y **Start_date** (fecha inicio de la venta de la propiedad) para buscar la diferencia de días entre la fecha de la creación del anuncio y el inicio de la operación de venta y con esos valores crear una nueva columna que indique cuando se tardo la propiedad en venderse. Con esta información podríamos analizar que tipos de propiedades y en qué barrios se vendieron más rápido las propiedades. Por sentido común, podríamos decir generalmente que si una propiedad se vende más rápido es porque es más barata, cuestión que habría que analizar.

Una última opción que nos hubiera gustado explorar es probar más modelos de clasificación y regresión, por ejemplo ensambles híbridos como stacking o voting, para analizar si estos tienen mejor performance que los elegidos.

Tiempo dedicado

Indicar brevemente en qué tarea trabajó cada integrante del equipo durante estas semanas.Si trabajaron en las mismas tareas lo detallan en cada caso (como en el ejemplo el armado de reporte). Deben indicar el promedio de horas semanales que dedicaron al trabajo práctico. En esta tabla solo deben incluir las tareas que realizaron luego de entregar el CHP1.

Integrante	Tarea	Prom. Hs Semana
Adriana Iglesias	Clasificación, Random Forest Regresión, XGBoot Regresión	6
Claudia Ramos	Armado de Reporte Clasificación, XGBoot Regresión	11
Ricardo Contreras	Clasificación, Gradient Boost regresión, XGBoot Regresión	7
Anita Vernieri	Clasificación KNN Regresión, XGBoot Regresión,	7