**Trabajo Práctico N°1 - Grupo 10**

**Análisis Exploratorio**

El conjunto de datos consta inicialmente de 20 columnas y 460154 entradas. Tras filtrar los datos que nos interesan para este TP, nos quedan 94249 entradas de interés.

Hay varios tipos de datos que nos interesan:

- tenemos datos cuantitativos, como las coordenadas o las propiedades, que permiten especificar cada dato y disponer de datos variados y de un conjunto de datos más completo,

- tenemos también datos cualitativos que permiten categorizar y especificar cada propiedad con diversas etiquetas.

**Preprocesamiento de Datos**

1. Eliminación de columnas

Decidimos eliminar las columnas "place\_l2", "operation" y "property\_currency", que no son útiles porque eran nuestros filtros iniciales y, por lo tanto, sólo tienen un valor en nuestro conjunto de datos. También eliminamos "place\_l5" y "place\_l6" porque estaban vacías.

Por último, decidimos eliminar "id", "property\_title", "start\_date", "end\_date" y "created\_on" porque sólo identifican una única venta y no un grupo de propiedades. No obstante, no estamos cerrados a reintegrar las columnas "start\_date", "end\_date" y/o "created\_on", que podrían ayudar a identificar tendencias mensuales en las ventas de propiedades.

1. Correlaciones

Hemos podido detectar las siguientes correlaciones:

- "property\_rooms" y "property\_bedrooms" : 0.86

- "property\_price" y "property\_rooms" : 0.48

- "property\_price" y "property\_bedrooms" : 0.42

- "property\_surface\_total" y "property\_surface\_covered" : 0.34

- "latitud" y "longitud" : -0.28

1. Nuevos features

Hemos creado una nueva variable, el precio por metro cuadrado por venta. Pudimos ver que existe una fuerte correlación entre el precio por metro cuadrado y el precio, lo que puede parecer lógico a pesar de la definición de esa nueva variable. Además, existe una pequeña correlación con los variables de latitud y longitud, lo que nos permite entender que el precio depende mucho del barrio en el que se sitúa.

1. Valores atípicos

Para analizar los valores atípicos, llevamos a cabo tres procesos diferentes.

En primer lugar, se comprobó la coherencia de los datos de acuerdo con las tres reglas siguientes:

- el número de habitaciones debe ser mayor o igual que el número de dormitorios

- la superficie cubierta no puede ser estrictamente superior a la superficie total

- cualquier valor negativo se tomaría como absoluto para tener sólo valores positivos.

Por lo tanto, hemos aumentado los valores de property\_bedrooms por property\_rooms y property\_surface\_covered por property\_surface\_total y hemos transformado todos los valores negativos.

A continuación, hemos realizado un análisis multivariante de property\_bedrooms, property\_rooms, property\_surface\_covered, property\_surface\_total y property\_price para identificar los valores atípicos y corregirlos mediante la imputación de regresión y la técnica MICE.

Por último, realizamos un análisis univariante de todas las variables para eliminar las que estaban demasiado alejadas del rango intercuartílico, y comprobamos que las latitudes y longitudes correspondían al barrio correspondiente.

1. Datos faltantes

Utilizamos tres técnicas diferentes para completar los datos que faltaban.

En primer lugar, para las variables "latitud", "longitud" y "place\_l3" (respectivamente 4,21%, 4,21% y 0,48% de datos faltantes), utilizamos un CSV que delimita geográficamente todos los barrios de Buenos Aires (<https://data.buenosaires.gob.ar/dataset/barrios>) para completar la variable "place\_l3" a partir de las coordenadas y las coordenadas a partir del centro del barrio según el valor de "place\_l3". Este método no es perfecto ni preciso, pero nos permite aumentar nuestros datos de manera coherente. Las entradas que se quedaron sin una de estas variables se suprimieron porque su número era reducido (en torno al 0,6%).

En segundo lugar, suprimimos la columna "place\_l4" (95,97% de datos ausentes) y sustituimos los de "place\_l3" por los de "place\_l4" si existían, ya que sólo contenían divisiones dentro de Palermo.

Por último, hemos utilizado el principio MICE para completar los datos que faltaban en las columnas "property\_rooms" (1,14%), "property\_bedrooms" (11,74%), "property\_surface\_total" (5,54%) y "property\_surface\_covered" (3,8%) a partir de las demás columnas utilizando el principio de regresión e intentando acercarnos lo más posible a la realidad.

**Visualizaciones**

**Une image contenant texte, capture d’écran, diagramme, Tracé

Description générée automatiquement**

Este gráfico de barras representa la diversidad de los barrios presentes en el conjunto de datos. Sin embargo, podemos ver que algunos barrios, como Palermo y Belgrano, están sobrerrepresentados con relación a los demás, lo que complicará la predicción para algunos barrios menos representados, como Liniers.

Une image contenant texte, capture d’écran, diagramme, ligne

Description générée automatiquement

Este diagrama de dispersión pone de manifiesto la correlación positiva entre el número de habitaciones y el número de dormitorios. También podemos ver que la distribución está bastante centrada en valores inferiores a 20, lo que parece coherente dadas las propiedades de Buenos Aires.

Puede que no necesitemos transformar los datos, pero sí eliminar aquellos que parezcan incoherentes (¡como los que tienen 70 dormitorios y sólo 4 habitaciones!). Por tanto, un criterio podría ser que el número de habitaciones sea siempre mayor o igual que el número de dormitorios.

**Clustering**

En primer lugar, analizamos la tendencia a la agrupación mediante el coeficiente de Hopkins, que se aproxima a 0, lo que nos permite afirmar que la agrupación de datos es posible en nuestro caso.

A continuación, estimamos la cantidad ideal de clusters mediante la regla del codo:

Une image contenant ligne, Tracé, diagramme, texte

Description générée automatiquement

Consideramos que el mejor número de clusters es 2. Esto se confirmó calculando el coeficiente de Silhouette, que mostró que 2 era el número ideal.

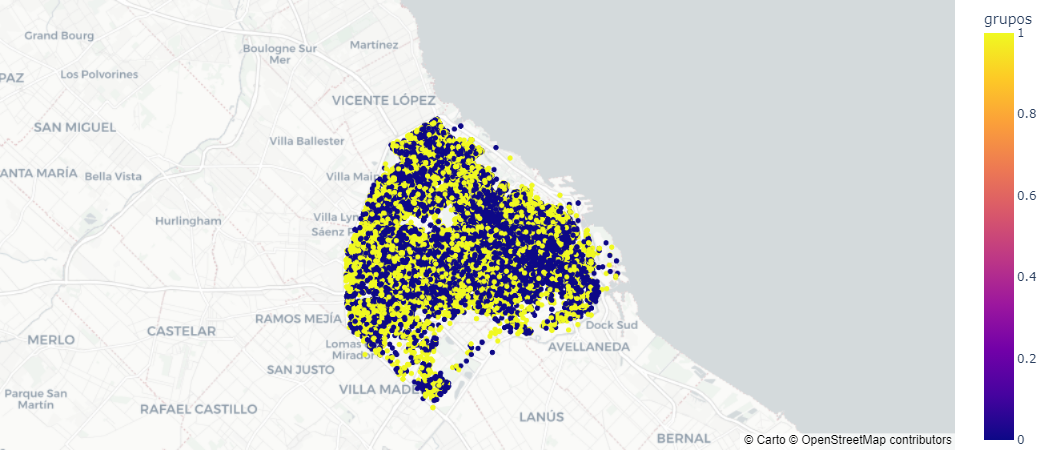
A continuación, realizamos un análisis de Silhouette para ver qué observaciones se habían clasificado erróneamente. Podemos observar que están presentes en pequeño número, lo que confirma la buena agrupación de los datos.

Une image contenant Tracé, ligne, diagramme, capture d’écran

Description générée automatiquement

Parece entenderse que en un grupo se incluyen las propiedades con muchas habitaciones y dormitorios, que tienen una mayor superficie y un precio más elevado (independientemente del precio por metro cuadrado), y que, por el contrario, en el segundo grupo se incluyen las propiedades más pequeñas, con menos habitaciones y un precio más bajo.

Finalmente, acá tiene un mapa que muestra todas las propiedades de Buenos Aires según su grupo:



**Clasificación**

* **Construcción del modelo**

Para todos los modelos, transformamos los datos de acuerdo con las siguientes reglas:

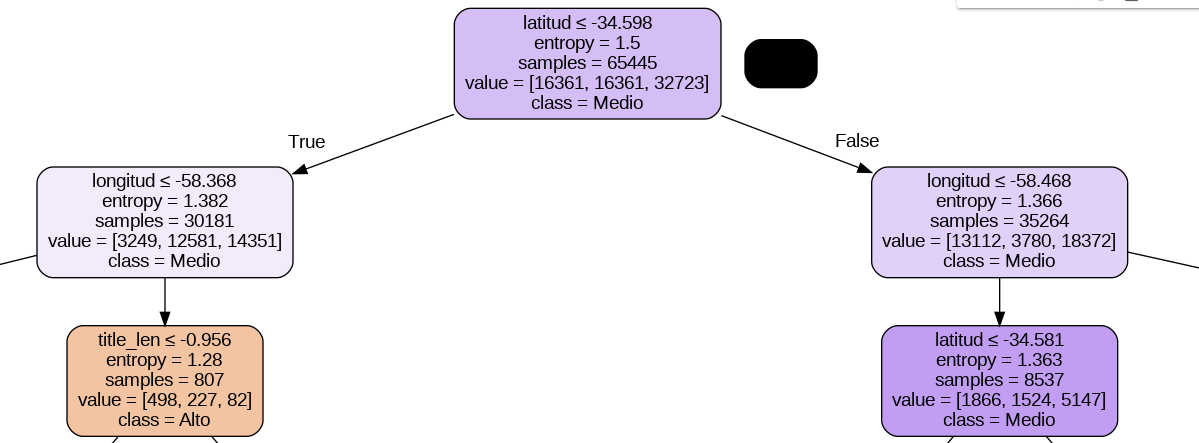
- las columnas numéricas se han normalizado utilizando un StandardScaler

- la columna place\_l3 se ha sustituido por dos columnas normalizadas que contienen el número de palabras y el número de caracteres del barrio

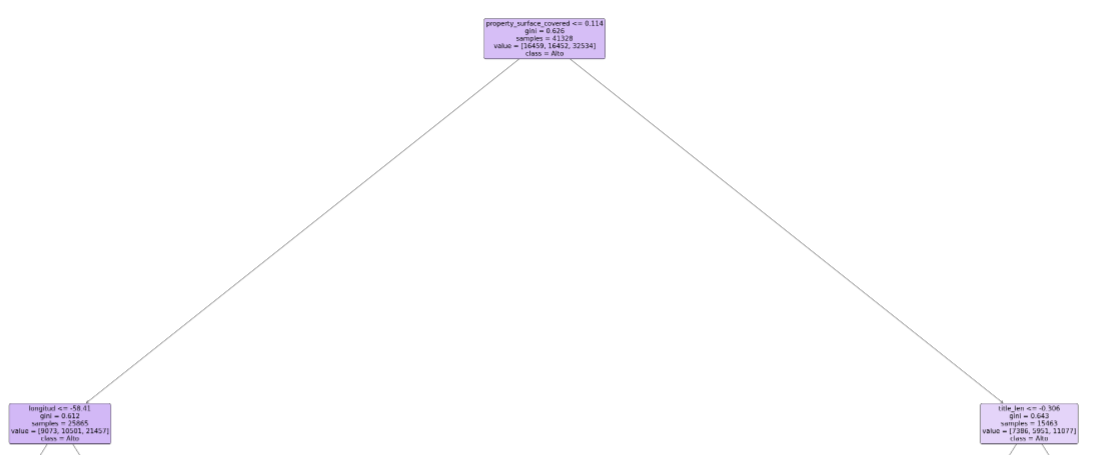
- la columna property\_type se ha desglosado mediante el método get\_dummies

Árbol de Decisión

* Optimizamos hiperparámetros, y utilizamos randomcv.best\_params\_ y randomcv.best\_score\_ con F1 Score
* Usamos K-fold Cross Validation y utilizamos folds=15
* De métrica usamos make\_scorer(f1\_score, average='micro')
* Imagen del árbol generado



Random Forest

* hiperparámetros usamos random\_cv\_random\_forest\_version.best\_params\_
* usamos folds = 5
* make\_scorer(f1\_score)

Modelo Regresion logistica

* Optimizamos hiperparámetros, y utilizamos LogisticRegressionCV con 10 folds.
* De métrica usamos el accuracy score, el recall score y el precisión 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
* XXX: si seleccionaron alguna métrica adicional…

Confeccionar el siguiente cuadro con esta información:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Modelo | F1-Test | Precision Test | Recall Test | Accuracy Test | XXX |
| Arbol de Decision |  |  |  |  |  |
| Random Forest |  |  |  |  |  |
| … |  |  |  |  |  |

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

**Nota: indicar brevemente en qué consiste cada modelo de la tabla, por ejemplo**

Arbol de decision: {'max\_depth': 7, 'criterion': 'entropy', 'ccp\_alpha': 0.02}

Random Forest: {'criterion': 'entropy', 'min\_samples\_leaf': 1, 'min\_samples\_split': 16, 'n\_estimators': 50}

* **Elección del modelo**

En base a los resultados obtenidos responder las siguientes preguntas

¿Qué modelo elegirían para clasificar el tipo de precio de las propiedades?¿ Por qué?

**Regresión**

Hemos realizado las mismas transformaciones que para la parte de clasificación.

* **Construcción del modelo**

KNN

* Hemos utilizado el K-fold Cross Validation con 5 folds y 10 iteraciones, porque como el conjunto de datos es bastante grande, es preferible no tener demasiado tiempo de calculo.
* La métrica utilizada para buscar los hiperparámetros es el R2 Score porque permite interpretar de manera más fácil las performances de un algoritmo de regresión.
* La performance de entrenamiento es de 0.8237, mientras que la performance de evaluación es de 0.0087, lo que no es tan bueno. El MSE de evaluación es de 80899641762.71526.

XGBoost

* Hemos utilizado el K-fold Cross Validation con 5 folds y 10 iteraciones, porque como el conjunto de datos es bastante grande, es preferible no tener demasiado tiempo de calculo.
* La métrica utilizada para buscar los hiperparámetros es el R2 Score porque permite interpretar de manera más fácil las performances de un algoritmo de regresión.
* La performance de entrenamiento es de 0.8210, mientras que la performance de evaluación es de 0.0655, lo que no es tan bueno. El MSE de evaluación es de 76261081426.09407.

Modelo a elección: Ridge Regression

* Hemos utilizado el K-fold Cross Validation con 5 folds y 10 iteraciones, porque como el conjunto de datos es bastante grande, es preferible no tener demasiado tiempo de calculo.
* La métrica utilizada para buscar los hiperparámetros es el R2 Score porque permite interpretar de manera más fácil las performances de un algoritmo de regresión.
* La performance de entrenamiento es de 0.6713, mientras que la performance de evaluación es de -0.3784, lo que no es bueno. El MSE de evaluación es de 112484482868.8376.
* **Cuadro de Resultados**

Medidas de rendimiento en el conjunto de TEST:

* MSE
* RMSE

|  |  |  |
| --- | --- | --- |
| Modelo | MSE | RMSE |
| KNN | 80899641762.71526 | 284428.623318 |
| XGBoost | 76261081426.09407 | 276154.090004 |
| Ridge Regression | 112484482868.8376 | 335387.064254 |

En todos los casos, los resultados de la evaluación son mucho peores que en el momento del entrenamiento.

* **Elección del modelo**

En nuestra opinión, el mejor modelo es el XGBoost porque parece obtener mejores resultados, tanto en el entrenamiento como en la evaluación.

**Conclusiones Finales**

Realizar las conclusiones correspondientes al trabajo realizado en su totalidad, destacando principalmente los aspectos que consideren más relevantes. Mencionar también conclusiones producto de la experimentación (que puede o no haber sido volcada en la entrega).

Comentar brevemente qué otras opciones hubiesen explorado y quedaron fuera del alcance de este trabajo: qué experimentos hubiesen querido realizar o qué técnicas hubiesen querido utilizar y no lo hicieron (por tiempo, complejidad, falta de teórica/práctica etc).

**Tiempo dedicado**

|  |  |  |
| --- | --- | --- |
| Integrante | Tarea | Prom. Hs Semana |
| Flavian Ferré | Clasificacion  Regresión  KNN,XGBoost  armado de reporte | 5 |
| Alan Richmond | Clasificacion  Entrenamiento y Prediccion  armado de reporte | 5 |
| Alan Mejia | armado de reporte |  |