UNIVERSIDAD DE BUENOS AIRES
FACULTAD DE INGENIERÍA

75.06 ORGANIZACIÓN DE DATOS

TRABAJO PRÁCTICO 2

MACHINE LEARNING

2C - 2019

Grupo 31 - Data Mundo Hiper Mega Red

Pablo De Giano	**95040**
Maximiliano Romero	**99118**
César Castellanos	**81404**

<hoja en blanco>

Objetivo **4**

Introducción **4**

Búsqueda de Features **5**

Algoritmos utilizados **9**

Ridge 9

Lasso 9

RandomForest 10

XGBoost 10

CATBoost 11

LightGBM 12

Conclusiones **12**

Objetivo

El objetivo del informe es poder explicar los criterios utilizados para la aplicación de los distintos algoritmos de Machine Learning para lograr estimar el valor de mercado de cada publicación de un inmueble, en el sitio de compra-venta y alquiler inmobiliario denominado zonaprop.com[1] en México.
El repositorio se encuentra en https://github.com/cessar5/7506_20192C

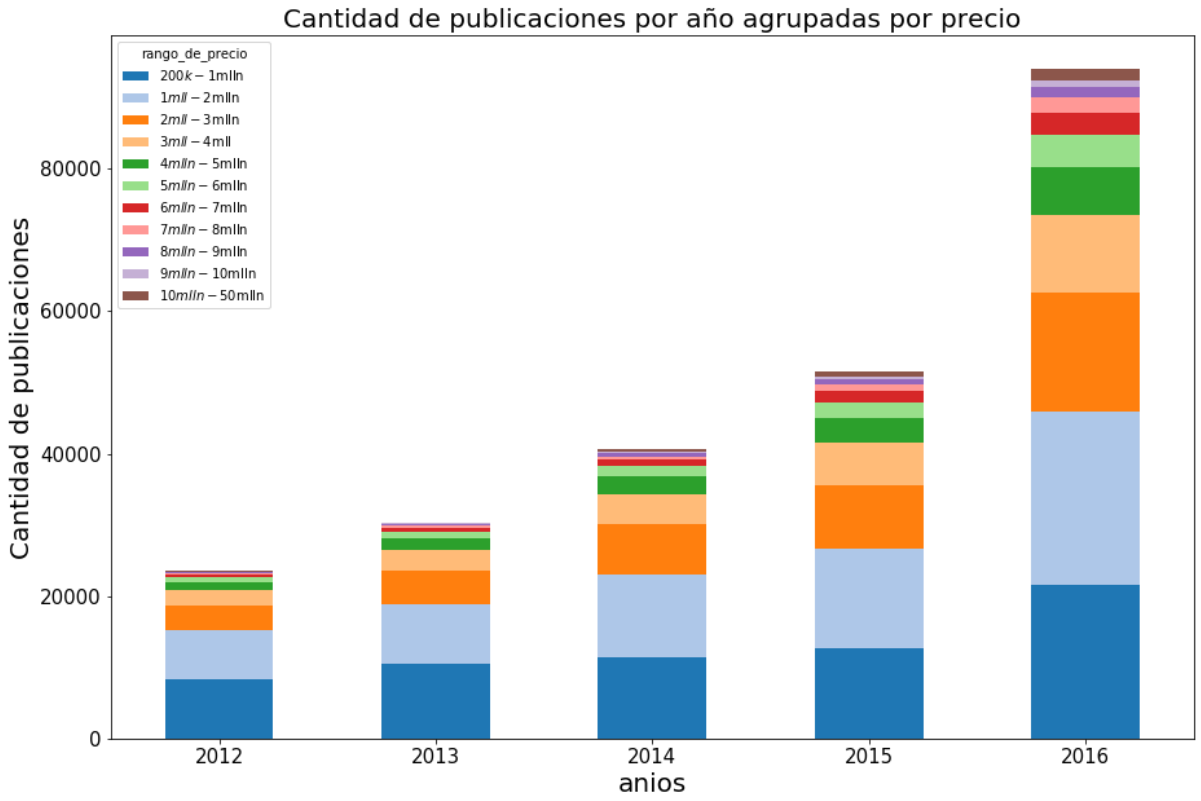
Introducción

Este trabajo práctico consiste en aplicar distintos algoritmos de *Machine Learning* para poder determinar, para cada id de publicación el precio del correspondiente inmueble en base a las distintas características del mismo.
Para luego poder participar de una competencia con los demás grupos presentes en la materia a través de la plataforma *Kaggle*.

El set de entrenamientos consiste en un solo dataset (*train.csv*) provisto por la empresa mencionada anteriormente, el cual contiene propiedades en venta entre los años 2012 y 2016, valuadas en pesos mexicanos. El mismo contra de 240.000 filas y 22 columnas.
Por otro lado se tiene el archivo *test.csv*, con 60.000 filas y 21 columnas, que contiene aquellas publicaciones a las que tenemos que predecir su valor.

Vemos como es la evolución del precio de las propiedades a través de los años de la publicaciones

[1] Sitio perteneciente a la empresa Navent.



Se observa que la mayoría de las publicaciones se ve entre 200 mil y 2 millones de pesos mexicanos.
Luego de realizar las predicciones del set de test obtuvimos resultados muy diferentes a los obtuvimos al momento de entrenar el set de test.
Entendemos tal vez que había outliers en el precio que influyan en el set de test.df

Búsqueda de Features

Para entrar en tema, empezaremos detallando los diferentes *features* obtenidos luego de realizar *feature engineering*. Es decir, aquellas características que consideramos que son relevantes con respecto a aquello que queremos estimar, y hacen que los algoritmos de machine learning puedan funcionar.

Para poder llevar a cabo dicha búsqueda, se tuvieron en consideración los resultados obtenidos de la realización del análisis exploratorio de los datos durante el trabajo práctico número 1, y sus respectivas conclusiones.

Los features utilizados fueron los siguientes:

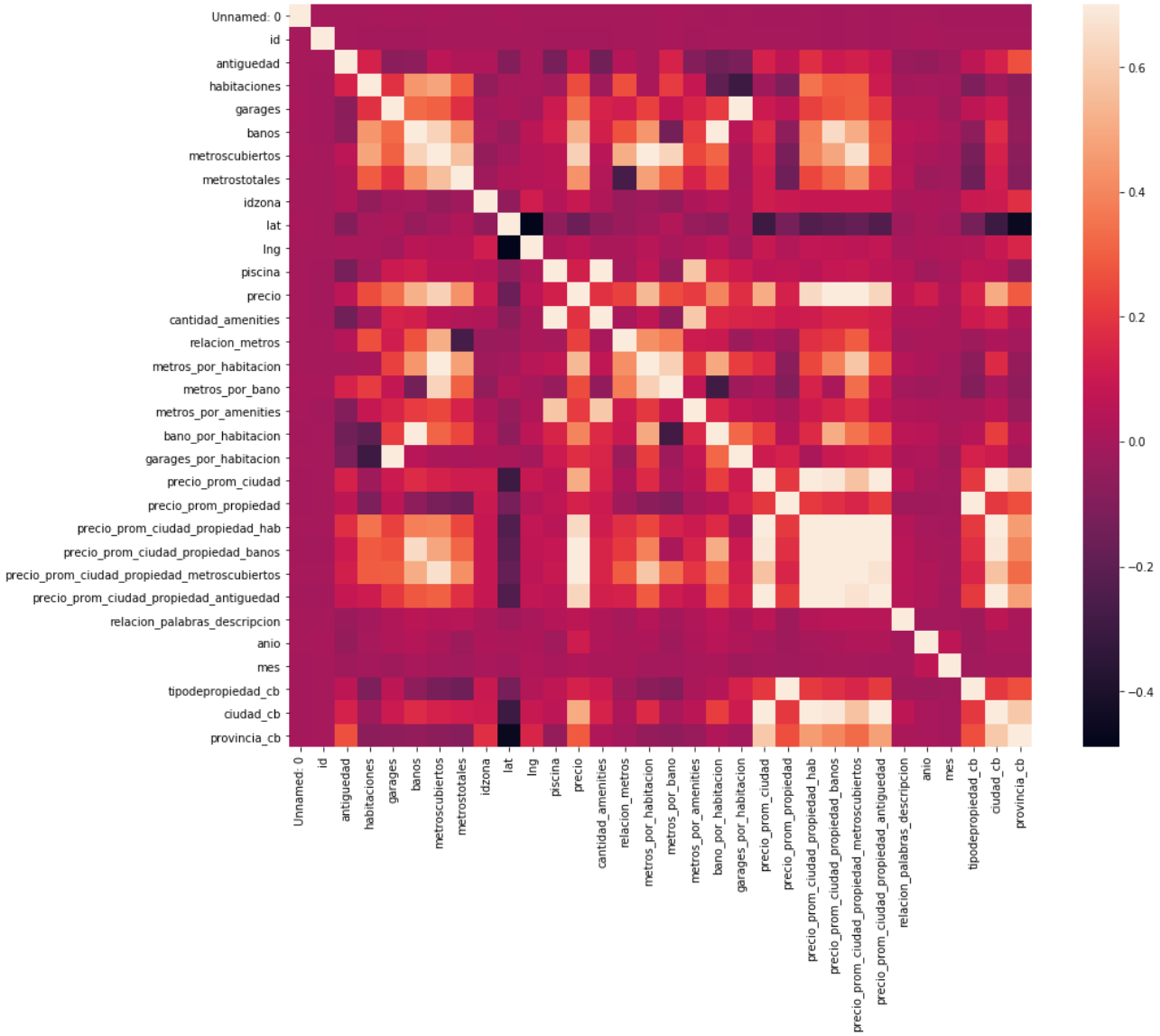
- Antigüedad: antigüedad de la propiedad.
- Habitaciones: cantidad de habitaciones.

- Garages: cantidad de garages.
- Baños: cantidad de baños.
- Metroscubiertos: metros cubiertos de la propiedad.
- Metrostotales: metros totales de la propiedad.
- Gimnasio: si el edificio o la propiedad tiene un gimnasio.
- Usosmultiples: si el edificio o la propiedad tiene un salón de usos múltiples (SUM).
- Piscina: si el edificio o la propiedad tiene un piscina.
- Escuelascercanas: si la propiedad tiene escuelas cerca.
- Centroscomercialescercanos: si la propiedad tiene comercios cerca.
- Cantidad_amenities: cantidad total de amenities *(gimnasio-usos múltiples-piscina)*.
- Cantidad_servicios: cantidad total de servicios *(escuelas-comercios cerca)*.
- Relacion_metros: relación entre los metros cubiertos sobre los metros totales *(cubiertos / totales)*.
- Metros_por_habitacion: metros cubiertos por habitación.
- Metros_por_bano: metros cubiertos por baño.
- Metros_por_amenities: metros cubiertos por cantidad total de amenities.
- Es_avenida: si en la dirección figura que el inmueble se encuentra en una avenida.
- Bano_por_habitacion: cantidad de baños por habitación.
- Garages_por_habitacion: cantidad de garages por habitación.
- Precio_prom_propiedad: precio promedio según el tipo de propiedad.
- Seguridad_descripcion: aparece la palabra seguridad (o derivados) en la descripción del inmueble.
- Moderno_descripcion: aparecen palabras relacionados con la modernidad en la descripción del inmueble.
- Hermoso_descripcion: en la descripción utiliza palabras positivas.
- Estado_descripcion: si en la descripción del inmueble hacen referencia a que la misma se encuentra en buen estado.
- Seguridad_titulo/Moderno_titulo/Hermoso_titulo/Estado_titulo: hacen referencia a los mismos puntos anteriores pero si las mismas palabras se encuentran en el título de la publicación.
- Relacion_palabras_descripcion: relación entre la cantidad de palabras mencionadas sobre el total de palabras en la descripción.
- Cantidad_palabras_descripcion: cantidad de palabras de los ítems anteriores en la descripción.
- Provincia_top5: si la propiedad pertenece

- Es_ciudad_capital: si la ciudad en la que se encuentra la propiedad es capital del estado.
- Ciudad_turistica_top15: si se encuentra en una de las 15 ciudades más turísticas de México.
- Precio_prom_ciudad: precio promedio según la ciudad.
- precio_prom_ciudad_propiedad_metroscubiertos: precio promedio según la ciudad, el tipo de propiedad y los metros cuadrados.
- precio_prom_ciudad_propiedad_banos: precio promedio según la ciudad, el tipo de propiedad y cantidad de baños de la propiedad.
- precio_prom_ciudad_propiedad_hab: precio promedio según la ciudad, el tipo de propiedad y cantidad de habitaciones.
- precio_prom_ciudad_propiedad_antiguedad: precio promedio según la ciudad, el tipo de propiedad y antigüedad del inmueble.
- Idzona: zona ubicación de la propiedad.
- Anio: año en el que se realizó la publicación de la propiedad.
- Mes: mes en que se realizó la publicación.

Además se realizó un encoded según el tipo de propiedad.

Luego de realizar el feature engineering generamos la matriz de correlación que nos permitirá entender cuál es la relación de la variables en particular cómo se relaciona el precio con respecto a las demás variables. Aquí vemos una fuerte relación con la variables que creamos en relación a los precios promedios en cuanto a ciudad propiedad y otro feature (metros cubiertos, baños, habitaciones)


Unnamed: 0
id
antiguedad
habitaciones
garages
banos
metroscubiertos
metrostotales
idzona
lat
lng
piscina
precio
cantidad_amenities
relacion_metros
metros_por_habitacion
metros_por_bano
metros_por_amenities
bano_por_habitacion
garages_por_habitacion
precio_prom_ciudad
precio_prom_propiedad
precio_prom_ciudad_propiedad_hab
precio_prom_ciudad_propiedad_banos
precio_prom_ciudad_propiedad_metroscubiertos
precio_prom_ciudad_propiedad_antiguedad
relacion_palabras_descripcion
anio
mes
tipodepropiedad_cb
ciudad_cb
provincia_cb
0.6
0.4
0.2
0.0
−0.2
−0.4

Algoritmos utilizados

En ésta sección mostraremos todos los modelos/algoritmos utilizados durante todo el proceso de entrenamiento de nuestro algoritmos de Machine Learning.
Algunos de estos modelos se usaron para poder llegar a mejores resultados en cuanto a las predicciones que tienen como objetivo el presente trabajo, mientras que otros, se utilizaron solamente para ver cómo se comportaba nuestro sistema.

Ridge

Ridge regression (Regresión Ridge o regularización de Tikhonov), es un modelo que resuelve un modelo de regresión donde la función de pérdida es la función lineal de cuadrados mínimos, y la regularización viene dada por la norma euclídea (L2).

Por otra parte dicho modelo puede mejorar los errores de predicción al reducir en tamaño los coeficientes de regresión que sean demasiado grandes para reducir el sobreajuste (*overfitting*), pero no realiza selección de variables y por tanto no produce un modelo más interpretable.

Subiendo este modelo a la plataforma de competición *Kaggle* obtuvimos como resultado un score de 1393380.69873. Siendo este unos de los primeros submit a la competencia en donde no usamos todos los parámetros encontrados hasta último momento debido a que es un método simple.

Lasso

Lasso (Least Absolute Shrinkage and Selection Operator) es un método de análisis de regresión que realiza selección de variables y regularización para mejorar la exactitud e interpretabilidad del modelo estadístico.

Es un modelo muy similar conceptualmente a la regresión Ridge, agregando una penalización para los coeficientes distintos de cero.
Pero a diferencia de Ridge que penaliza la suma de los coeficientes cuadrados, Lasso penaliza la suma de sus valores absolutos.
Con dicho modelo los resultados obtenidos oscilaron entre 1529422.98423 y 1344851.18403.

RandomForest

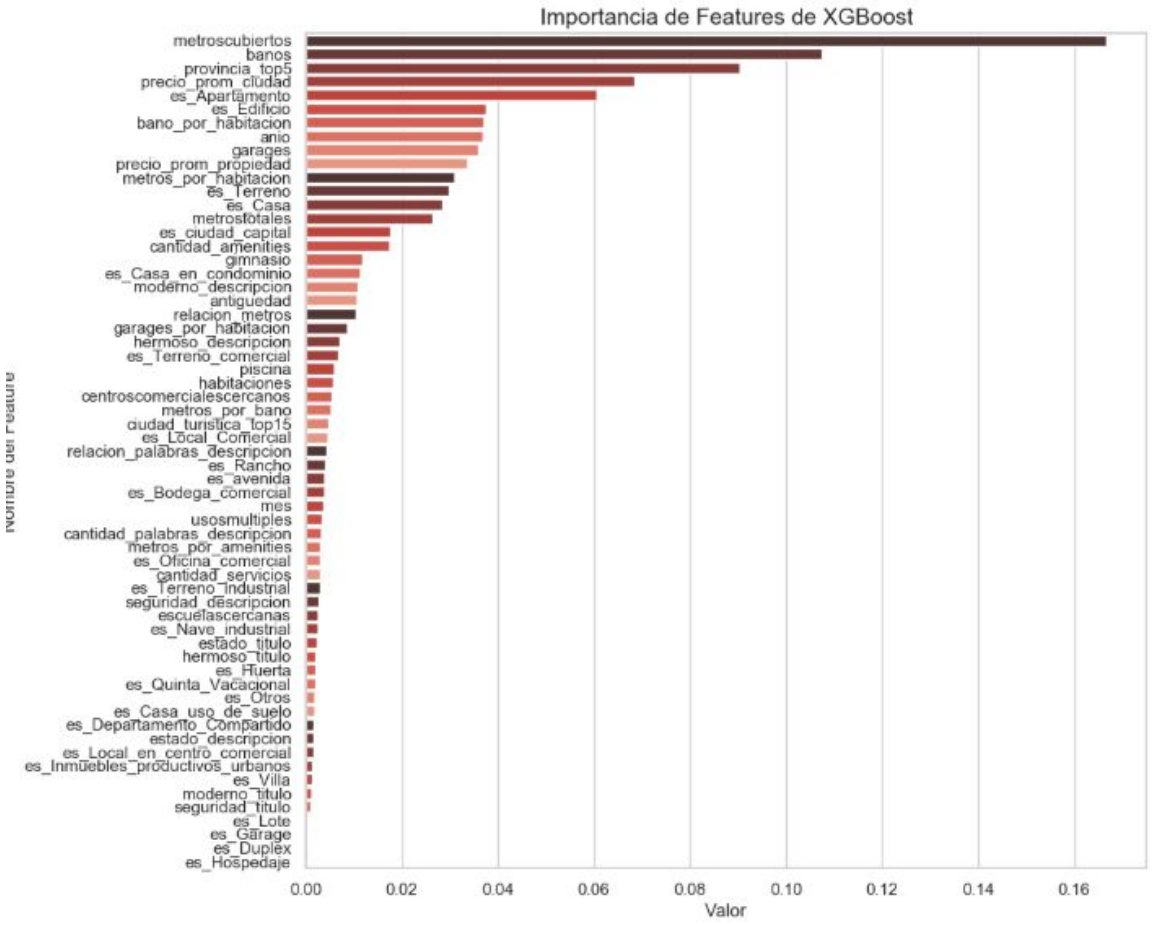
Es una combinación de árboles predictores tal que cada árbol depende de los valores de un vector aleatorio probado independientemente y con la misma distribución para cada uno de estos. En Random Forest se ejecutan varios algoritmos de árbol de decisiones en lugar de uno solo. Para clasificar un nuevo objeto basado en atributos, cada árbol de decisión da una clasificación y finalmente la decisión con mayor “votos” es la predicción del algoritmo. En un árbol de decisión, una entrada se introduce en la parte superior y hacia abajo a medida que atraviesa el árbol de los datos se acumulan en conjuntos cada vez más pequeños.
Si bien el modelo permite la optimización de hiperparametros, utilizando RandomSearchCV, no se logró mejorar los resultados obtenidos tal que que hagan una diferencia con el modelo base. Aquí surge el inconveniente de al intentar aumentar los estimadores generó mejores resultados con este modelo pero a su vez fue un limitante ya que no se pudo probar con más de 120 estimadores porque generaba problemas con el consumo de cpu.
Con dicho modelo los resultados obtenidos oscilaron entre 594802.11561y 2529478.47723.

XGBoost

XGBoost es una implementación de código abierto popular y eficiente del algoritmo de árboles aumentados de gradientes.
La potenciación de gradientes es un algoritmo de aprendizaje supervisado que intenta predecir de forma apropiada una variable de destino mediante la combinación de estimaciones de un conjunto de modelos más simples y más débiles.

Simplificando, podemos decir que XGBoost está netamente basado en el estado del arte en clasificación, y realiza un Boosting de árboles de decisión.

Uno de los pros de XGBoost es que permite la optimización de hiperparámetros y la posibilidad de visualizar los mejores features del modelo en cuestión. Realizando ésta última acción obtuvimos el siguiente gráfico:



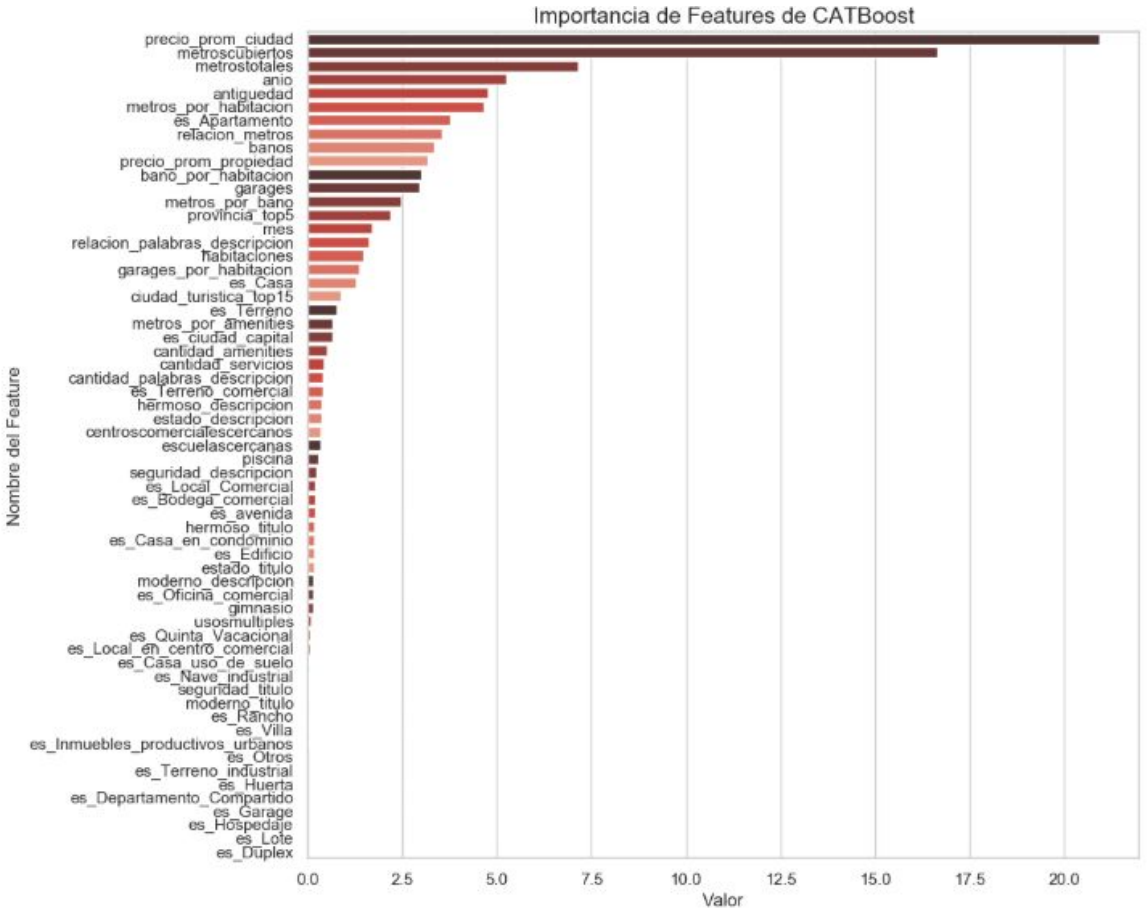
Con este modelo los resultados en la competencia variaron entre 1532683.69450 y 577980.12206.

CATBoost

El nombre *CatBoost* proviene de la unión de los términos "*Category*" y "*Boosting*". *Category* hace referencia al hecho de que la librería funciona perfectamente con múltiples categorías de datos, como audio, texto e imagen, incluidos datos históricos. Es un algoritmo de aprendizaje automático basado en potenciación del gradiente ("*Gradient boosting*") desarrollado por los investigadores de Yandex que es adecuado en múltiples aplicaciones.

La potenciación del gradiente es una técnica que se aplica en múltiples tipos de problemas como la detección de fraude, motores de recomendaciones y predicciones. Además de obtener buenos resultados con relativamente menos datos los modelos de Deep Learning.

Los resultados en la competencia utilizando este modelo fueron desde 556959.54912 hasta 824756.87687.



La figura anterior ilustra la importancia que le da CATBoost a los distintos features utilizados.

LightGBM

*LightGBM e*s un marco de potenciación del gradiente rápido, distribuido y de alto rendimiento, basado en algoritmos de árbol de decisión. Se utiliza para la categorización, la clasificación y muchas otras tareas de aprendizaje automático.
Las principales ventajas de dicho modelo son: soporte de aprendizaje paralelo y GPU, menor uso de memoria, capaz de entrenar datos a gran escala, entre otras. Estas surgen, principalmente, de que LightGBM utiliza algoritmos basados en histogramas, que agrupan los valores de los features continuos en contenedores discretos.

Se utilizó RandomSearchCV para estimar hiperparametros que se adapten a nuestro modelo de manera eficiente y se ajustó manualmente a partir de los mismos hasta encontrar los valores óptimos.

Los resultado obtenidos en la competencia con este modelo nos dieron valores entre 532012.21935 y 573846.94092.

Conclusiones

El informe trata de ilustrar todo el procedimiento realizado en la elaboración del trabajo práctico número 2 de la materia Organización de Datos, así como también los resultados o scores obtenidos.

El mejor resultado obtenido fue de 532012.21935. El mismo fue generado a partir de LightGBM, con CatBoost Encoded para los features categóricos “tipodepropiedad”, “provincia”, “ciudad” y utilizando los 30 features más importantes obtenidos previamente según CatBoost.