**Diplomatura en Ciencia de Datos, Aprendizaje Automático y sus Aplicaciones**

**Informe Trabajo Práctico 4 Mentoría**:

“Análisis del mercado inmobiliario de la Ciudad de Buenos Aires 2017”

**Mentor:** Javier Lezama

**Integrantes Grupo:**

-Navarro Agustina

-Calle Manuel

-Tagle Gabriel

**Agosto 2019**

Introducción

Para la confección del informe se desarrolló una notebook en Jupyter donde se aplicaron los conocimientos de la materia de la diplomatura: “Aprendizaje Automático Supervisado”.

En la notebook usamos diferentes librerías que nos permitieron combinar herramientas de manejo de grandes volúmenes de datos, análisis estadísticos, división de datos, y algoritmos de predicción. Algunas de las librerías nuevas que hemos utilizado son las de sklearn model selection, metrics, tree y neighbors.

El mentor de este grupo es Javier Lezama quien nos orientó en lo relacionado al contenido del dataset y a la descripción del caso para poder desarrollar este informe.

División de datos en conjuntos de entrenamiento y evaluación.

Para comenzar el análisis, se plantearon las variables que son importantes para poder estimar la variable precio de la propiedad en dólares. Se definieron las siguientes:

* Superficie Total en m2 ('surface\_total\_in\_m2')
* Codigo de Barrio ('barrio\_cod')
* Tipo de Propiedad ('property\_type')
* Cantidad de Habitaciones ('rooms')
* Superficie Cubierta en m2 ('surface\_covered\_in\_m2')

Aplicando “train\_test\_split” de la librería “sklearn.model\_selection” se dividió el conjunto de datos en 2 grupos. Uno de validación y uno de entrenamiento, con el 20% y 80% del total del dataset respectivamente. Y a su vez el conjunto de datos se divide en las features que harán la predicción y la columna a predecir, es decir la columna *target,* que contiene el precio aproximado en usd (‘price\_aprox\_usd’).

Elección de un modelo de Regresión: Árboles de Decisión

El modelo de regresión que aplicaremos es el de árboles de decisión: DecisionTreeRegressor. Este modelo aplica un árbol para seleccionar los valores de las features que se aproximan más al valor a predecir. Lo que realizan estos algoritmos es tomar decisiones en base a los valores de las features analizadas y así generar distintas ramas para llegar al valor a predecir en la regresión.

Para la ejecución de este modelo se deben definir los hiper parámetros correspondientes tales como el criterio de división de features, la cantidad máxima de niveles de profundidad del árbol, la forma de dividir los datos, cantidad de hojas, etc. Para realizar esta selección se aplicó la técnica GridSearchCV, en la cual se prueba un rango de parámetros y este método nos devuelve la mejor combinación.

Selección de Hiper parámetros

Cómo se mencionó previamente, para la selección de hiper parámetros se aplicó la técnica de GridSearchCV, que permite hacer una búsqueda exhaustiva de varios parámetros para obtener el mejor resultado de un modelo de aprendizaje automático. Para la elección de los Hiper parámetros se definieron rangos de posibles valores para el máximo de hojas, de divisiones y niveles de profundidad, siendo valores discretos naturales. Dentro de los criterios y formas de división, se armó un array de todos los valores posibles de cada uno.

Una vez definidos los rangos de parámetros posibles, se entrena el modelo y se obtienen los mejores hiper parámetros probados. En este caso el mejor conjunto fue el siguiente:

*Mejor conjunto de parámetros:*

*{'criterion': 'mse', 'max\_depth': 8, 'min\_samples\_leaf': 1, 'min\_samples\_split': 5, 'splitter': 'random'}*

Este resultado, se mide en base a una función de costo, que es la que nos indica cuán eficientes son los parámetros y la confianza del modelo. En este caso se utilizó la función R2 o coeficiente de determinación, en la que valores cercanos al 1, indican el mejor score obtenido.

Luego del entrenamiento del modelo y la elección de los hiper parámetros, verificamos el score en base a la función de costo R2 en el conjunto de datos de train y de test, los valores obtenidos son los siguientes:

*Mejor R2 Score Train: 0.9646334276168972*

*Mejor R2 Score Validación: 0.938783618017728*

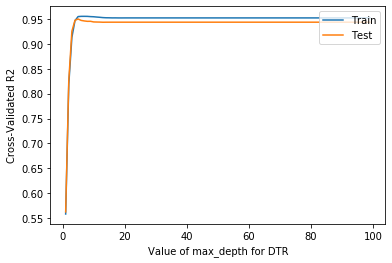
Por lo que vemos de resultado, es bastante bueno el modelo de regresión y no se nota un overfitting en los datos de train, ya que las medidas en ambos conjuntos de datos son aproximadas.

Métricas sobre el conjunto de evaluación

En este punto analizaremos la variación de la función de costo según la variabilidad de los hiperparametros. Para ello se corre el modelo n veces, como parámetros queramos probar y también se realiza una validación cruzada de los mismos para obtener los scores resultantes.

*Max\_depth*

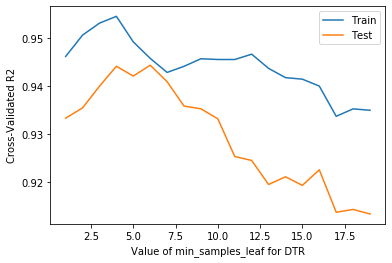
En primer lugar se probó el parámetro de la *máxima cantidad de niveles de profundidad (max\_depth)* del árbol, con un rango de 0 a 100 valores y este fue el resultado en los datos de Train y Test:



Este gráfico, acompaña al mejor resultado obtenido por el método GridSearchCv, donde nos decía que el mejor valor del parámetro era: *'max\_depth': 8*

*Min\_samples\_leaf:*

Si tenemos en cuenta la mínima cantidad de hojas del árbol, se probaron 20 valores enteros:

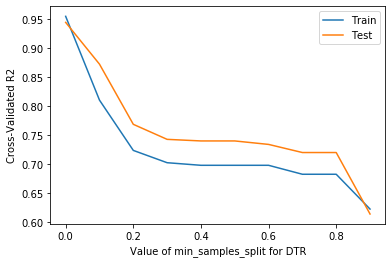


Este gráfico nos indica la diferencia de comportamiento entre los dos set de datos y vemos como la tendencia al aumento de cantidad de hojas hace que el error entre train y test sea cada vez mayor, esto nos muestra el fenómeno de overfitting del modelo en los datos de train.

Viendo el gráfico podemos decir que el mejor parámetro sería un valor cercano a 5, ya que es el máximo valor de la curva, donde el error en ambos conjuntos es menor, y la métrica r2 es cercana a 1. Recordando el parámetro seleccionado por el GridSearchCv: *min\_samples\_leaf: 4*

*Min\_samples\_split:*

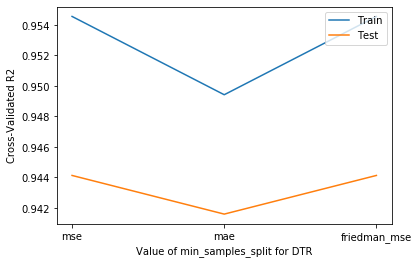
Teniendo en cuenta este parámetro se probaron todos los valores posibles, obteniendo como resultado el siguiente gráfico:



Como vemos el mejor valor es el parámetro cercano a 0 para ambos casos, ya que a medida que aumenta disminuye es score, aumentando el error en el modelo.

*Criterion*

Si evaluamos el Criterio del árbol obtenemos el siguiente gráfico:



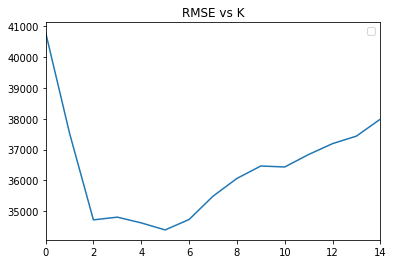
Lo cual nos determina que el mejor parámetro es el MSE, al igual que nos determinó el GridSearchCV.

Elección de un modelo de Regresión: KNN

En este caso se analizará el problema con un modelo de regresión de vecinos más cercanos, buscando predecir los valores de nuestra columna Target: precio aproximado en usd.

En primer lugar se entrena el modelo con los valores por defecto de todos los hiperparametros, menos el de la cantidad de vecinos cercanos que se tendrán en cuenta para realizar la regresión: *n\_neighbors:(1-15).*

Para esta evaluación del parámetro vamos a utilizar la función de costo: RMSE (error cuadrático medio) y con esto obtuvimos el siguiente resultado:



Graficando los distintos valores de RMSE para cada K, se puede observar que se minimiza el RMSE con K=5:

Luego se aplicó una búsqueda exhaustiva para optimizar los demás parámetros y verificar si obtenemos que el K óptimo es igual a 5 como nos indica el gráfico anterior. Gracias a GRidSerchCV obtuvimos que el mejor conjunto de parámetros es:

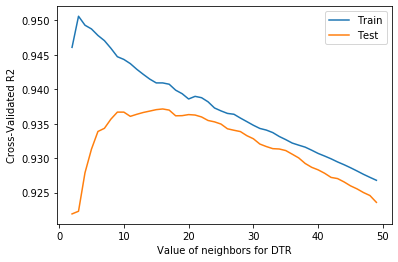
*Mejores Parámetros en Train: {'algorithm': 'ball\_tree', 'leaf\_size': 5, 'n\_neighbors': 3, 'weights': 'distance'}*

*Mejores Parámetros en Test: {'algorithm': 'brute', 'leaf\_size': 1, 'n\_neighbors': 5, 'weights': 'distance'}*

*Mejor Score R2 Train: 0.999031755931401*

*Mejor Score R2 Test: 0.9579490977275466*

Por otro lado también comparamos el parámetro de la cantidad de vecinos con la función de costo R2 y obtuvimos el siguiente resultado:



En ambos casos el valor óptimo es entre 5 y 10, un valor que se adecue a ambos conjuntos de datos, evitando así el overfitting en el conjunto de training y pudiendo predecir correctamente en el conjunto de test.

Conclusiones

Este trabajo práctico permitió al grupo aplicar conocimientos necesarios para entender el concepto del aprendizaje automático y la elección de un modelo que permita predecir los precios de las propiedades buscando el mayor grado de exactitud.

En este trabajo se exploraron modelos de regresión por árboles de decisión y K vecinos más cercanos, para elegir el mejor modelo y luego analizar los hiper parámetros que nos den un mejor score garantizándonos la confianza del modelo, analizándolo en los conjuntos de datos de entrenamiento y validación.

La elección de un modelo de regresión para predecir datos, puede ser una forma de encontrar rápidamente un modelo simple que pueda predecir con exactitud nuestra variable objetivo del set de datos. En este caso, los modelos analizados fueron DecisionTree y KNN. Y de ellos el que mejor confianza nos dió fue el KNN.

Para sacar esta conclusión se analizaron los posibles rangos de parametros posibles y se evaluaron las métricas de RMSE y R2.

Sin embargo, aún se puede seguir indagando en mejorar la predicción pensando en desafiar más algunos parámetros como, validar si fueron acertadas las variables del set de datos utilizadas o limpieza de outliers antes de aplicar el modelo elegido.

A lo largo de los siguientes prácticos se irán aplicando nuevos conceptos hasta finalmente optimizar el modelo de predicción para poder obtener la mayor precisión con el menor costo computacional.