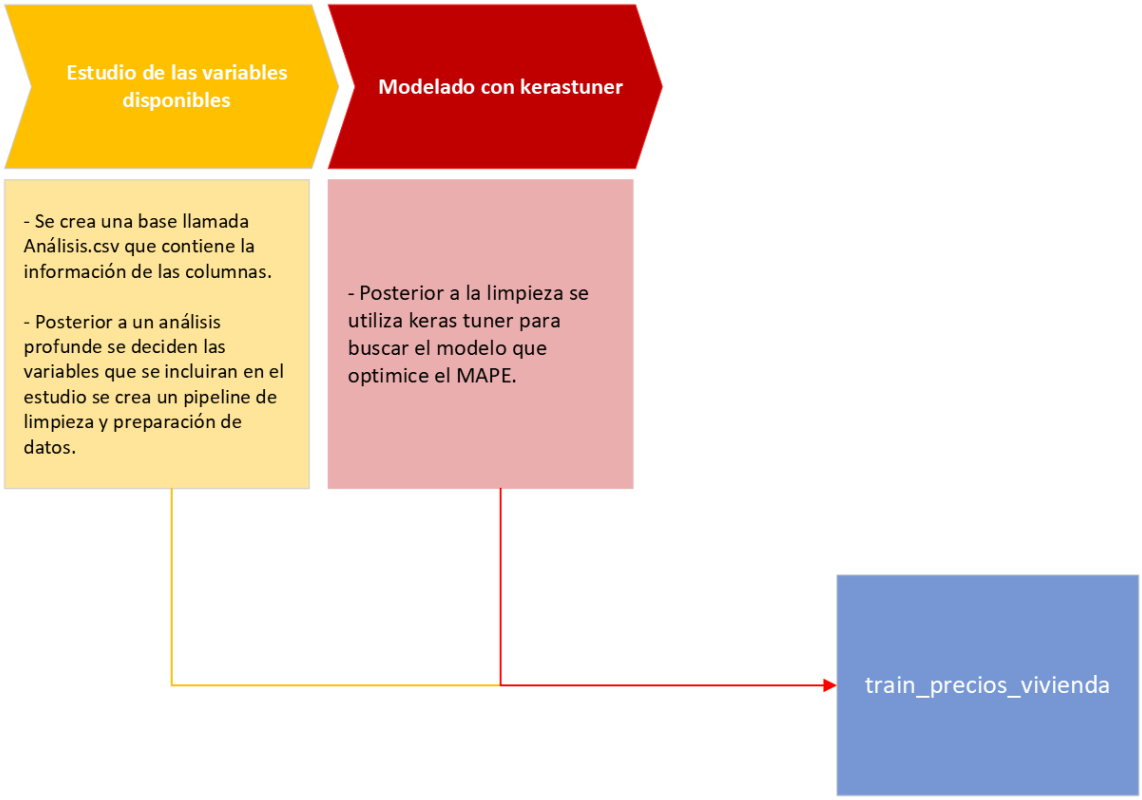
Predecir precios de vivienda

Introducción

En este documento se resume el ejercicio para crear un modelo que permita predecir el precio de la vivienda según varias características. La base entregada consta con 11699 registros y 222 columnas. La característica más peculiar de esta base es que requiere varios procesos de limpieza. Se sugiere el siguiente proceso:



Toda la documentación del ejercicio se guardó en el repositorio abierto: https://github.com/Izainea/pruebas_vivi/tree/master/prueba_2_Analitica_Predictiva_cient%C3%ADfico_vivi utilizando una estructura documental similar a la presentada en la metodología de proyectos de ciencia de datos para equipos de Microsoft. Dicha estructura identifica tres carpetas principales del proyecto:

- **Código** donde encontrarán dos cuadernos:
    - *Exploración.ipynb* que contiene el proceso completo en el cual se define la base Análisis que se utilizo para definer las variables a utilizar.

- **Datos** donde se guarda toda la información que se obtuvo de este ejercicio, si bien, por la sencillez del proyecto no se generó una carpeta para datos crudos, intermedios y procesados.

Se mantuvo los nombres acordes con lo obtenido en cada fase del proyecto. Se destacan las siguientes bases:

- *Análisis.csv* Contiene información resumida de las columnas. Creada con Python.
- *Análisis_2.csv* Contiene información resumida de las columnas, aquí ya se aplicaron varios análisis para determinar las variables adecuadas para el ejercicio.
- *Train_precios_vivienda.csv* Información completa que se utilizó para el ejercicio.

- **Documentación** Aloja únicamente este documento.

Limpieza de información

Después del análisis se identifica la naturaleza de las variables y se limpian de acuerdo a su naturaleza. Se trata de automatizar este proceso haciendo una diferencia entre variables booleanas, variables numéricas y variables categóricas. En este ejercicio se elimina la información incoherente.

Búsqueda del modelo

Para esta implementación se intenta trabajar con kerastuner, se proponen tres capas que pueden tener entre 2 y 216 neuronas. Se configura para que trabaje con el error MAPE. No obstante los resultados no son buenos. El mejor modelo tiene las siguientes carcaterísticas:

Model: "sequential" _____
Layer (type) Output Shape Param #
===

dense (Dense) (8652, 212) 5595104

dense_1 (Dense) (8652, 12) 2556

dense_2 (Dense) (8652, 154) 2002

dense_3 (Dense) (8652, 1) 155
===

Total params: 5,599,817 Trainable params: 5,599,817 Non-trainable params: 0
_____

Cuando se calcula el MAPE para la base de testeo se obtiene 232.89.

Conclusiones

El error que aquí se presenta está ocasionado por una falta de rigurosidad en el tratamiento de datos. No obstante, frente a diversos ejercicios mejoró mucho en contraste con los primeros resultados.