Conjunto de problemas 3: ¿Ganar dinero con el ML?

"¡¡¡Todo es cuestión de ubicación, ubicación, ubicación!!!"

MECA 4107

Fecha de entrega: 26 de julio a las 23:59 en Bloque Neon

1 Introducción

El fiasco de Zillow inspiró este conjunto de problemas.[1] Zillow desarrolló algoritmos para comprar casas. Sin embargo, sus modelos sobrestimaron considerablemente el precio de las viviendas. Esta sobreestimación supuso unas pérdidas de unos 500 millones para la empresa y una reducción aproximada del 25% de su plantilla.

En este conjunto de problemas, trataremos de evitar(?) un fiasco similar al comprar casas en Chapinero, Bogotà; y en El Poblado, Medell'ın.

El conjunto de datos para este conjunto de problemas procede de https://www.properati.com.co. Los datos contienen información sobre los precios de los listados, así como características de las propiedades en venta. Los datos están disponibles en el archivo data.zip y contienen tres archivos: datos de entrenamiento, datos de prueba y una plantilla de envío. Debe enviar sus predicciones para los datos de prueba siguiendo la plantilla de envío.

1.1 Instrucciones generales

El objetivo principal es construir un modelo predictivo de los precios de venta. A partir del histórico trabajo de Rosen "Hedonic Prices and Implicit Markets: Product Differentiation in Pure Competition" (1974), sabemos que un vector de sus características, $C = (c_1, c_2, \ldots, c_n)$, describe un bien diferenciado.

En el caso de una vivienda, estas características pueden incluir atributos estructurales (por ejemplo, el número de habitaciones), servicios públicos del barrio (por ejemplo, la calidad de la escuela local) y servicios ambientales locales (por ejemplo, la calidad del aire). Así, podemos escribir el precio de mercado de la vivienda como

$$P_i = f(c_{i1}, c_{i2}, \ldots, c_{in})$$

Sin embargo, la teoría de Rosen no nos dice mucho sobre la forma funcional de f. En este conjunto de problemas, usted explorará diferentes modelos para obtener la mejor predicción posible.

[1]Para más información, consulte el siguiente artículo.

Hay dos resultados previstos:

1. Un documento .pdf.
2. Y un archivo .csv con las predicciones.

El documento no debe tener más de 3 (tres) páginas e incluir como máximo 6 anexos (tablas y/o figuras). Se puede añadir un apéndice, pero el documento principal debe ser autónomo. En concreto, un lector debe ser capaz de seguir el análisis del documento y convencerse de que es correcto y coherente sólo a partir del texto principal, sin consultar el apéndice.

El documento debe contener las siguientes secciones:

- Introducción. Aproveche esta sección para "vender" su modelo predictivo, mostrando las ventajas y desventajas del modelo elegido y el rendimiento esperado.
- Datos. Además de las variables incluidas en el conjunto de datos, aquí se requiere ampliarlo (recuerde ampliar los datos de entrenamiento y de prueba), como mínimo tiene que añadir cuatro variables adicionales:
  - Al menos 2 de estos modelos deben incluir predictores procedentes de fuentes externas; ambos pueden ser de callejeros abiertos.
  - Al menos 2 predictores procedentes del título o la descripción de las propiedades.

  Como siempre, trate esta sección como una oportunidad para presentar una narrativa convincente para justificar o defender sus elecciones de datos. Guíe al lector a través de su razonamiento sobre cómo ha "limpiado" los datos y los ha ampliado con datos adicionales. Descríbalo con estadísticas descriptivas, gráficos, etc. Como mínimo, debe incluir:
  - Una tabla con estadísticas descriptivas
  - Dos mapas (uno para cada ciudad principal), puedes elegir qué información mostrar

  Utilice sus conocimientos profesionales para añadir valor a esta sección. No la presente como una lista "seca" de ingredientes.
- Modelo y resultados. Cuando presente su modelo predictivo incluya:
  - Una explicación de las variables utilizadas para entrenar este modelo, recuerde utilizar las variables que añadió en la sección anterior.
  - Una explicación detallada sobre cómo se entrenó, la selección de los hiperparámetros y cualquier otra información relevante sobre el modelo.
  - Una discusión de su medida de evaluación.
- Conclusiones y recomendaciones. En esta sección se exponen brevemente las

principales conclusiones del trabajo.

2 Directrices adicionales

Espero las siguientes cosas del conjunto de problemas, la omisión de cualquiera de estas pautas será penalizada.

- Entregue su documento y un archivo .csv de presentación con las predicciones en bloque nebn.
  - Un ejemplo de cómo debe ser el archivo de presentación está en la carpeta de datos: submis- sion_template. csv. Este archivo incluye 2 columnas, una con la variable que identifica una propiedad y otra con su precio previsto.
  - Voy a juzgar las predicciones en base a los que gastan menos dinero y pueden comprar más propiedades. Es decir, si se sobrevalora una propiedad que se añade a su cuenta, si se infravalora, se ahorra. Sin embargo, si el precio predicho está subestimado por más de 40mill. COP (aprox. 10 mil dólares), será penalizado. En este caso, la venta no se llevaría a cabo, es decir, no podrá comprar el inmueble.
  - Por favor, siga la siguiente convención para el nombre de su archivo .csv. El nombre del archivo debe incluir el nombre predicciones, seguido de los apellidos de tus compañeros de equipo, y el número de variables que utilizaste para la predicción, todo ello separado por puntos suspensivos, por ejemplo, predicciones_gomez_matinez_sarmiento. csv
  - Asignaré puntos de bonificación en función de las clasificaciones relativas.
- Las tablas, las figuras y la redacción deben ser lo más nítidas posible. Etiquete y describa todas las variables, estadísticas, etc., incluidas en sus figuras y tablas.
- El documento debe apuntar e incluir un enlace a su repositorio de GitHub.
- Siga la plantilla del repositorio.
- El repositorio debería incluir un archivo README. Un buen LÉEME ayuda a que su proyecto se destaque de otros proyectos y es el primer archivo que una persona ve cuando se encuentra con su repositorio. Por lo tanto, este archivo debe ser lo suficientemente detallado como para centrarse en tu proyecto y en cómo lo hace, pero no tan largo como para perder la atención del lector. Por ejemplo, Project Awesome tiene una lista curada de READMEs interesantes.
- El repositorio debe tener al menos cinco (5) contribuciones sustanciales de cada miembro del equipo.
- Tu código debe ser legible e incluir comentarios. En la codificación, como en la escritura, un buen estilo de codificación es fundamental para que el código sea legible. Te animo a seguir la guía de estilo de Tidyverse.