Ciencia de Datos – Trabajo Práctico

Locales en venta en CABA

Martins, Steven
Zimmerman, Michel
Vartorelli, Nicolás

1 Introducción y objetivos

1.1 Introducción

A continuación, en el siguiente documento se explicará en detalle el trabajo desarrollado por el grupo durante el segundo cuatrimestre del año 2021.

Dicha investigación se encuentra enmarcada bajo la cursada de Ciencia de Datos, materia electiva del último año de Ingeniería Industrial en la Universidad Tecnológica Nacional.

A lo largo de los últimos meses, hemos aprendido diversos conceptos dentro de la programación, tales como Machine Learning, Ciencia de datos y Análisis de Datos.

El objetivo del trabajo es plasmar dichos conocimientos en un caso práctico real. Para esto se utilizó Jupyter Notebook, el cual posibilitó la creación de documentos donde se puede utilizar distintos lenguajes de programación, así como también facilitó a la conexión de fuentes internas como externas de datos.

Dentro de los tantos lenguajes de programación existentes, se utilizó el lenguaje de código abierto denominado Python, siendo este uno de los lenguajes más utilizados hoy en día a nivel mundial por los desarrolladores.

Este lenguaje posee un gran número de librerías creadas por desarrolladores, lo que permite potenciar el trabajo y ahorrar tiempo a partir de evitar el arduo trabajo de codificar ciertas funciones de manera manual. Lo que se traduce, a fin de cuentas, en un ahorro del tiempo para llegar a obtener los resultados finales.

1.2 Objetivo

El objetivo de este trabajo es predecir el precio de un local en la Ciudad de Buenos Aires a partir de la data disponible en el portal de información del Gobierno de la Ciudad para el 2020. Si se deseara, se podrían actualizar los valores al corriente año realizando un ajuste por inflación y parámetros similares. De todas maneras, esto queda fuera del foco de análisis de este trabajo.

2 Descripción del dataset

2.1 Origen de la base de datos

La base de datos utilizada para la realización del presente trabajo fue obtenida del sitio web oficial del Gobierno de la Ciudad Autónoma de Buenos Aires.

Desde hace aproximadamente una década, la Ciudad de Buenos Aires, desarrolló esta sección con diversas bases de datos con el fin de que cualquier ciudadano pueda acceder a la información y con ella realizar los análisis que deseen.

El link utilizado para descargar el dataset es el siguiente (en su versión 2020):
https://data.buenosaires.gob.ar/dataset/locales-en-venta

2.2 Profundización del Dataset

Se descargó un dataset en formato comma-separated values (csv) que posee los locales en ventas dentro de la Ciudad de Buenos Aires.

En primer lugar, la base de datos tuvo que ser limpiada para luego poder realizar el análisis correcto de la información vía modelos de Machine Learning.

El dataset poseía información de 6575 locales. Con el limpiado que se realizó sobre la base, se perdieron algunos locales, por ejemplo, debido a que no había información relevante sobre los mismos.

Luego se descartaron aquellas columnas con datos irrelevantes para el análisis y finalmente se realizaron verificaciones, como, por ejemplo, chequear que no hubiera valores "N/A" en las filas, que el formato de cada variable sea el correcto y conteos de valores.

3 Análisis exploratorio de datos

Como se mencionó anteriormente, el dataset posee 6575 filas con información sobre locales en ventas.

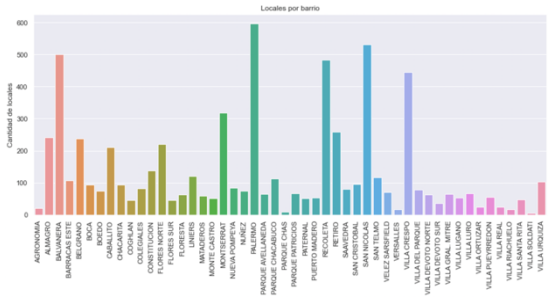
Se descartaron las columnas irrelevantes y luego se analizó el contenido por columnas.

Se observó que no aparecían columnas con valores faltantes, a excepción de la columna "galería" que poseía un gran número de valores vacíos. Se asumió que aquellos locales con valores vacíos en galería eran debido a que el local no se encontraba dentro de una.

Por otro lado, en esta columna había discrepancia entre los datos, ciertas filas poseían "S1" en vez de "SI", por ende, se asumió que fue un error en la base y ambos valores correspondían a lo mismo

Gráficos

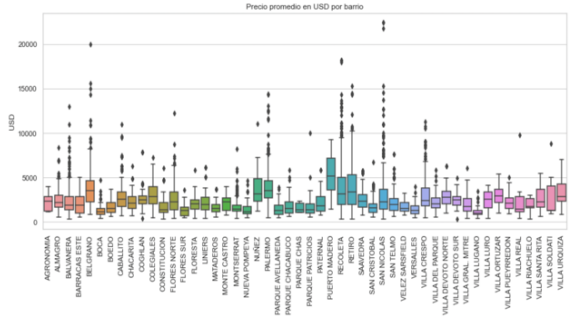
A continuación, se observa un gráfico con la cantidad de locales por barrio:



En el gráfico se observa que Palermo es la zona con más locales en venta, con aproximadamente 600 publicaciones. Esto tiene sentido a partir de ser una zona muy comercial, con gran cantidad de locales y con muchos negocios cerrando debido a la situación económica. Luego siguen San Telmo, Balvanera, Recoleta, San Nicolás y Villa Crespo con cerca de 500 locales por barrio.

Siguen Retiro y Montserrat con 300 locales y por último, el resto de los barrios con alrededor de 100 locales en venta por barrio.

El siguiente gráfico muestra los precios promedio, en dólares, por barrio:



Se destaca a San Nicolás y Puerto Madero como los barrios más caro de la Ciudad de Buenos Aires, algo que intuitivamente podía llegar a pensarse

debido a que son barrios muy turísticos, con gran cantidad de hoteles de lujo donde se hospedan empresarios y turistas del exterior y donde muchas multinacionales poseen sus oficinas administrativas.
Siguen Belgrano, Palermo, Recoleta, Retiro como los barrios con mayor costo.
Se puede observar que los barrios ubicados más al norte de la ciudad y además son los barrios más turísticos y con mayor cantidad de oficinas radicas, son los de un mayor costo.

4 Materiales y métodos

4.1 Métodos

Considerando la base de datos utilizada y teniendo en cuenta que la predicción se hará sobre valores continuos queda determinado que esto se trata de un problema de regresión. Utilizaremos aprendizaje supervisado, aprovechando el conocimiento de las etiquetas. Nuestro target será el "preciousd" o el precio de las propiedades en dólares.
En primera instancia, se separó el dataset en una parte de etiquetas "y", y una parte de variables independientes "x". Luego, ambas partes fueron distribuidas en un 20% a un set de testeo y 80% a un set de entrenamiento. La parte de training será la utilizada por los distintos modelos para determinar una función "óptima" que represente una relación entre las variables independientes y el target.

4.2 Estandarización y variables categóricas

Al haber variables categóricas como "barrios" y "galería" debemos convertirlas a valores que puedan ser utilizados por los modelos para intentar encontrar las funciones mencionadas. Para esto, utilizados Pandas, convertimos estas variables categóricas a variables numéricas binarias.
Luego, para evitar la distorsión natural de la data por tratarse esta de valores en rangos muy diferentes (precios de los locales vs metros cuadrados, por ejemplo) empleados un mecanismo de estandarización con media 0 y desvío estándar 1:

$$z = (x - u) / s$$

donde u es la media, s el desvío estándar y x el valor original.

4.3 Modelos de Machine Learning

Se entrenaron cuatro modelos de regresión:

- Linear Regression
- K-Nearest Neighbours Regressor
- Support Vector Regressor
- Random Forest Regressor

4.4 Optimización de hiper-parámetros

Para optimizar los hiper-parámetros se utilizó el método de Grid Search Cross Validation para todos los modelos a excepción de Linear Regression.
En el caso de KNN Regressor se intentó optimizar los siguientes parámetros: n_neighbors, algorithm. Se probaron distintos rangos y algoritmos y los ganadores fueron 3 neighbors y el algoritmo ball_tree.
Para Support Vector Regressor se iteró sobre: kernel, c, Gamma, coef0, shrinking. El kernel ganador fue el polinómico y el parámetro de regularización C ganador fue 16.
Para Random Forest se iteró sobre n_estimators, max_features, min_samples_leaf. Los parámetros

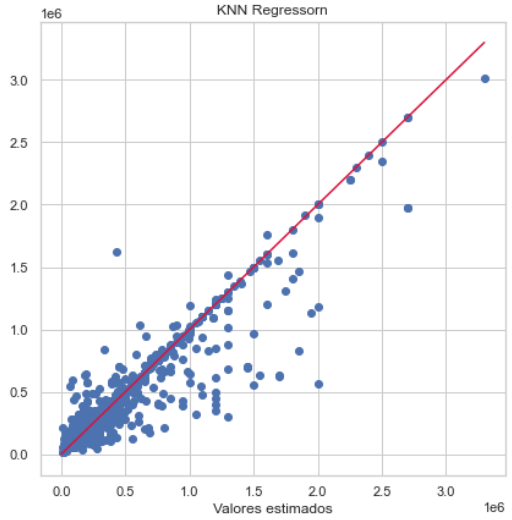
ganadores fueron: 0.7, 1 y 210, respectivamente.

5 Experimentos y resultados

5.1 KNN Regressor

Este regresor implementa el aprendizaje basándose en los neighbors más cercanos de cada punto, donde n_neighbors es un valor entero especificado por nosotros.

El algoritmo ganador "ball tree" divide de forma recursiva los datos en nodos definidos por un centroide y un radio, de modo que cada punto del nodo se encuentra dentro de la hiperesfera definida por y.



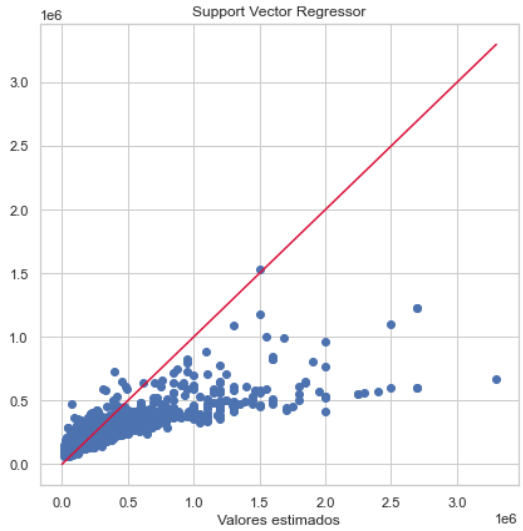
Este fue el algoritmo ganador entre los entrenados con un error cuadrático medio de 148604.4018.

5.2 Support Vector Regression

Busca maximizar el margen entre clases construyendo una función lineal. Es decir, determina un margen/radio como función de costo y trata de que todas las muestras estén dentro del margen. El hiper-parámetro es una función que penaliza muestras fuera del margen.

$$C\sum_{n=1}^{N}\xi_n + 1/2\,\|w\|^2$$

El resultado que nos dio es el siguiente:



Error cuadrático medio: 321987.8978

5.3 Linear Regression

Es una función lineal que se construye calculando parámetros "Beta" asociados a cada dimensión/feature.

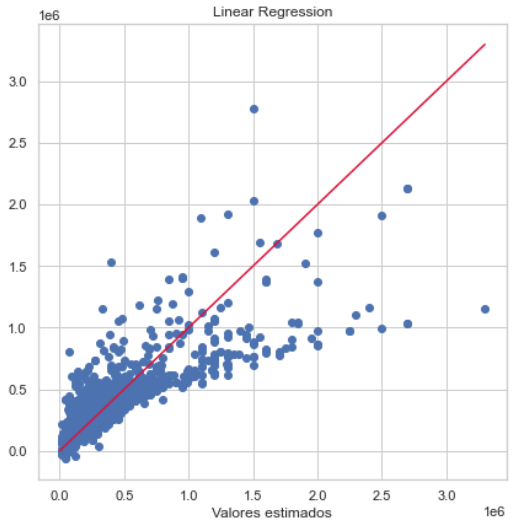
$$\widehat{y} = f\,(x, \beta)$$

$$\hat{y}(x, w) = w_0 + w_1x_1 + w_2x_2 + ... + w_px_p$$

Para obtener los valores de los parámetros del modelo utilizamos mínimos cuadrados ordinarios y obtenemos una única solución resolviendo:

$$\min_{\beta} \|X_w - y\|^2 \longrightarrow \hat{\beta} = (X^TX)^{-1}X^Ty$$

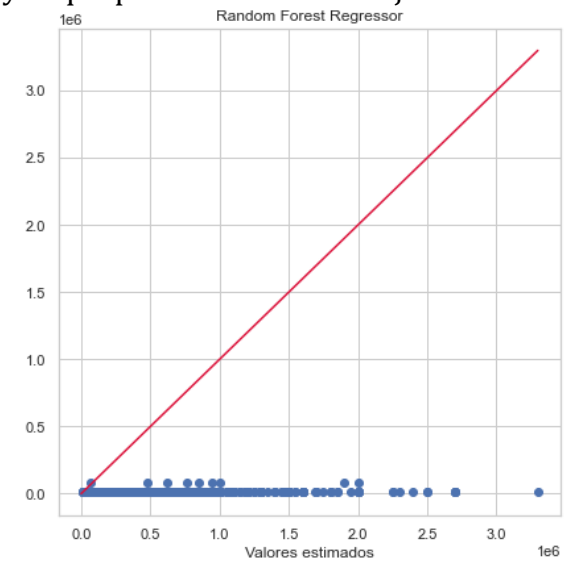
El resultado que nos dio es el siguiente:



Error cuadrático medio: 244039.1251

5.4 Random Forest Regressor

En modelo se crea un conjunto diverso de clasificadores mediante la introducción de aleatoriedad en la construcción del clasificador. La predicción final se da como la predicción promedio de los clasificadores individuales. Este es el algoritmo más demandante computacionalmente entre los elegidos y el que peor resultados arrojó.



Error cuadrático medio: 548466.4417

6 Discusión y conclusiones

Como podemos observar, el regresor K-Nearest Neighbors fue el que mejor logró representar la data con la que se trabajó. Inferimos que esto se debe a que trabaja mejor con modelos no-lineales y que tengan pocas features como es el caso de este dataset.

El mismo dio un error cuadrático medio 40% más pequeño que el que obtuvimos del segundo mejor modelo que es Regresión Lineal.

Con la información que nos brindan estas variables, podemos decir, que el modelo de aprendizaje supervisado de KNN es el que mejor va a poder estimar el valor en dólares de los locales de Capital Federal. Igualmente, se podrían plantear escenarios de mejora donde se podría reducir aún más los outliers en los precios, aumentando la cantidad de información histórica (agregando años previos), o intentando adquirir más features de otras fuentes.

7 Materiales y referencia

- Python Data Science Handbook (Autor: Jake VanderPlas)
- Python Aplicaciones Prácticas (Autor: Nolaso Valenzuela, Jorge Santiago)
- Python para finanzas quant (Autor: Juan Pablo Pissano)
- Scikit-learn: Machine Learning in Python (author={Pedregosa, F. and Varoquaux, G. and Gramfort, A. and Michel, V. and Thirion, B. and Grisel, O. and Blondel, M. and Prettenhofer, P. and Weiss, R. and Dubourg, V. and Vanderplas, J. and Passos, A. and Cournapeau, D. and Brucher)