Ciencia de Datos – TP Grupal

Valor de Propiedades - Regresión

Integrantes:

- Bosano, Ariel
- Innocenti G., Gianluca
- Ruiz Sánchez, Santiago

1 INTRODUCCIÓN

En el presente documento se procederá a explicar y detallar todo lo realizado por nuestro grupo para el desarrollo del trabajo práctico llevado a cabo. Esto significó varias jornadas de trabajo.

Dicho trabajo se encuentra enmarcado en la cursada de la materia Ciencia de Datos, la cual tiene carácter de electiva y es dictada en la Universidad Tecnológica Nacional (UTN) regional Buenos Aires.

El grupo se encuentra compuesto por 3 alumnos de dicha facultad, ambos estudiantes de la carrera Ingeniería Industrial.

Durante el transcurso de los últimos 4 meses hemos desarrollado los conceptos de Machine Learning, Ciencia de Datos y Análisis de Datos, los cuales se intentarán plasmar aquí.

Para ello se utilizó la herramienta **Jupyter Notebook** que consiste en un ambiente interactivo para la creación de documentos Jupyter, en los cuales se pueden utilizar variados lenguajes de programación para cualquier fin, así como también conectarse a fuentes externas o internas de datos.

El lenguaje de programación utilizado es **Phyton**, uno de los más utilizados actual y mundialmente por desarrolladores. Su gran fortaleza es la posibilidad de utilizar librerías desarrolladas por otras personas, potenciando el trabajo propio y también evitando la codificación y escritura de funciones (con una reducción en el tiempo requerido para obtener un resultado)

2 DESCRIPCIÓN DEL DATASET

Origen

El Dataset escogido fue obtenido del portal **Buenos Aires Data (1)** donde se canalizan todos los archivos del Gobierno de la Ciudad.

Este portal de datos nació en 2012 con el objetivo de facilitar el acceso a los ciudadanos a un catálogo de datos públicos y abiertos que la Ciudad produce diariamente.

Adicionalmente la web incluye datasets de terceros pero que se encuentran referidos a la ciudad de Buenos Aires, como algunos de movilidad o vivienda.

(1) https://data.buenosaires.gob.ar/

El origen propiamente de los datos es de un proveedor tercero, **Zonaprop**, página de venta y alquileres de propiedades muy concurridas por los Argentinos. Para conocer dicho origen, se realizó una consulta al portal de datos donde se confirmó que los datos no eran recogidos por la oficina del Gobierno de la Ciudad.

Dado que se trata de un dataset de una página de la industria inmobiliaria, el mismo posee información de todas las **propiedades en venta** que registró la compañía por propietarios o inmobiliarias.

Descripción

Si bien pueden encontrarse muchas versiones referidas a esta temática, se escogieron dos dataset distintos que se encuentran divididos por año:

(1) *Departamentos-en-Venta-2015.csv*
(2) *Departamentos-en-Venta-2016.csv*

Ambos fueron descargados en formato .csv, único formato ofrecido por el portal.

Cabe destacar que ambos fueron los más actualizados que disponía en su momento el Gobierno de la Ciudad de Buenos Aires y que, durante la realización del trabajo, todos los datasets fueron eliminados. Puede suponerse que no se seguirá actualizando esta información.

Objetivo

El objetivo de este proyecto es el de generar un modelo de regresión que sea capaz de predecir el valor de una propiedad, que se encuentre dentro de CABA, en base a un cierto conjunto de features.

3 ANÁLISIS EXPLORATORIO DE DATOS

Departamentos en Venta 2015

El dataset que contiene las propiedades publicadas en 2015 posee **11414 samples** y **35 features,** sin contar índices.

De las 11414 samples pudo verse que no había ninguna feature vacía, salvo en los casos de **Latitud y Longitud** en los que si se encontraron 621 casos vacíos. Dado que la Latitud y Longitud son samples muy dependientes, puede afirmarse que esas 621 faltantes en cada feature corresponden a los mismos samples.

Departamentos en Venta 2016

A diferencia del anterior, el Dataset del 2016 proviene del origen con más de 6 columnas presentes de vacíos (NaNs) en sus samples.

Se destacan las columnas **COMUNA, LATITUD, LONGITUD, BARRIO y CODIGO POSTAL**, con una presencia de 1382 NaNs sobre un total de 7562 samples.

Data Cleaning & Engineering

Para llevar a cabo un proceso de regresión, se procedió a hacer un **merge entre ambos datasets**.

Y a su vez para ello se realizaron varios pasos:

(1) Hallar la intersección entre las columnas de cada data set.

(2) Hacer foco en aquellas que se encuentran en la intersección. En estas se encontraron tres casos particulares en los cuales se hizo un análisis más exhaustivo (Barrio Normalizado / Barrio, Baños / Baño, Av_score / Nota)

(3) En base al análisis realizado en el punto anterior se hizo un cambio de nombre de columnas en los casos que correspondía.

Luego de realizar el merge, se contaba con un dataset de **18978 samples y 16 features**. Al cual se le hizo un merge adicional con un archivo realizado manualmente.

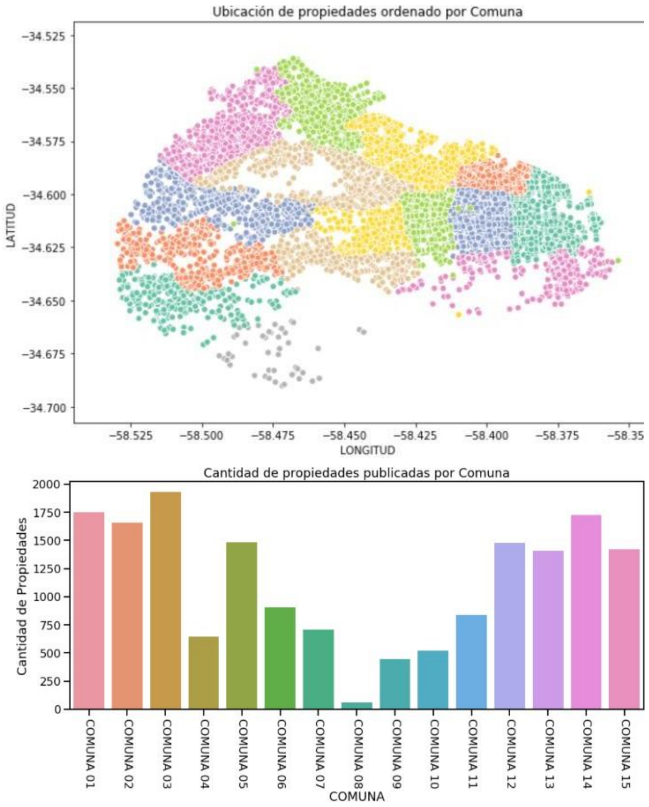
Este archivo fue creado con el propósito de **transformar los datos categóricos de Barrio** (nombre de barrio, Ej: PALERMO – BALVANERA) a **datos numéricos.**

Por último, de eliminaron los NaNs y Nulls provenientes de ambos datasets que no fueron filtrados previamente y se normalizó el feature **COMUNA** necesariamente debido a la diferencia en criterios entre ambos datasets.

Gráficos

Como podemos ver en los gráficos que aparecerán a continuación, la zona sur de la Capital Federal es donde se presenta la menor

cantidad de publicaciones de propiedades, esto se debe a que por la falta de inversión en infraestructura y el gran índice de inseguridad, no haya interés en el mercado inmobiliario por esa zona (esto se puede ver en el gráfico de precio promedio por comuna), generando que las empresas constructoras no construyan dejando una gran numero de terrenos sin edificar.



4 MATERIALES Y METODOS

En base a los datos que tenemos y al tipo de información que queremos predecir, decidimos utilizar aprendizaje supervisado. Por la naturaleza de este, vamos a tener una etiqueta para cada sample, que es una variable dependiente, y las distintas features que serían las variables independientes. Estas variables están relacionadas a través de una función y=f(x) que es desconocida, pero nosotros gracias a este tipo de aprendizaje vamos a aprender una función que se acerque lo más posible a la verdadera f(x) que posiblemente nunca conozcamos.

Para este caso particular, vamos a querer predecir el precio del M2 en función de ciertas características de las propiedades como son los M2, los ambientes, la antigüedad de la casa, los baños, el barrio y la comuna donde se ubican.

Se comenzó separando el dataset en dos, una parte la llamamos Y (etiquetas) y la otra parte X (variables independientes). Luego, procedimos a separar ambas partes en dos porciones una de entrenamiento y otra de prueba utilizando una herramienta de Sklearn; la porción de entrenamiento es utilizada por el modelo para aprender y generar la función correctamente y la porción de prueba se usa para poder verificar si esa función está generalizando adecuadamente y tiene un grado de precisión aceptable. Cuando realizamos esta separación, hay que evitar que se produzca “Overfitting” que suele estar asociado a modelos más complejos que se ajustan muy bien a los datos de train aunque generalizan mal para datos futuros o “Underfitting” que suele estar asociado a modelos muy sencillos que no logran ajustarse a la complejidad de los datos, nosotros consideramos que lo correcto sería destinar un 40% de los samples del dataset original como prueba y un 60% para entrenamiento.

Como muchas de las features tienen rangos muy distintos, por ejemplo, ambientes o antigüedad, utilizamos auto-scaling (el cual asume que cada una de ellas de manera individual responde a una distribución de probabilidad normal) para estandarizar los valores afectándolos por la media y el desvió estándar.

$$x'_i = \frac{(x_i - \mu)}{\sigma}$$

Cada feature después de pre-procesarla quedara con una media = 0 y un desvió estándar = 1.

El objetivo de todo lo que realizamos anteriormente es para preparar los datos para ser utilizados por los distintos modelos de

aprendizaje supervisado de regresión ya que nuestra variable Y es del tipo continuo.

Los principales modelos son:

1. Regresión Lineal
2. KNN
3. Support Vector Regression

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

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

$$\hat{y}(x, w) = w_0 + w_1 x_1 + w_2 x_2 + ... + w_p x_p$$

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

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

En cambio, en el entrenamiento de *KNN* se determinan K vecinos mas cercanos por distancia euclídea. El Yi para predecir se determina por la interpolación de los Y en los K vecinos y los pesos indican como se interpolarán cada K vecino.

$$d(x_a, x_b) = \sqrt{(x_{a1} - x_{b1})^2 + (x_{a2} - x_{b2})^2 + ... + (x_{ap} - x_{bp})^2}$$

Por último, *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-parametro es una función que penaliza muestras fuera del margen.

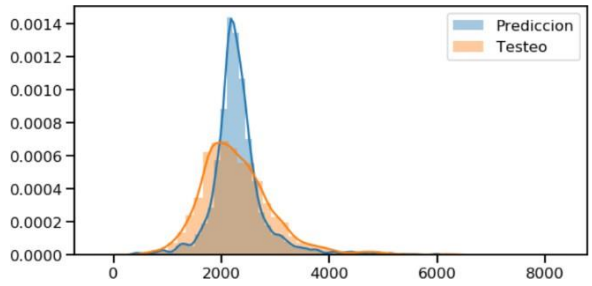
$$C \sum_{n=1}^{N} \xi_n + 1/2 \|w\|^2$$

Para este caso particular, nosotros decidimos utilizar Regresión Lineal y KNN.

5 RESULTADOS

Regresión Lineal

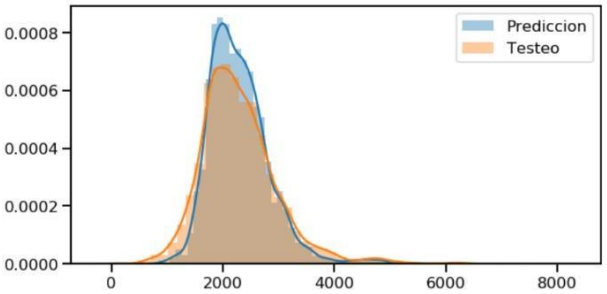
Aplicando este modelo, pudimos obtener un error cuadrático medio de 368.02, un coeficiente de determinación (R2) de 0.73 (el 73% de las variaciones de las variables dependientes esta explicada por la variación de las independientes) y tuvo la siguiente distribución:



KNN

Antes de aplicar el modelo, decidimos utilizar Grid Search que nos ayudara a seleccionar el mejor hiper-parametro, el cual fue k igual a 20 vecinos.

El modelo nos dio un error cuadrático medio de 286.20, un coeficiente de determinación (R2) de 0.84 y tuvo la siguiente distribución:



6 DISCUSION Y CONCLUSIONES

A partir de los resultados obtenidos en el punto 5, podemos determinar que la relación entre las X del dataset es más bien no lineal y que por ejemplo en el caso de aplicar Regresión Lineal por ser un clasificador del tipo lineal podríamos estar haciendo underfitting.

KNN como se nutre de las “no-linealidades” fue el que mejores resultados dio en comparación con el modelo6. El mismo dio un error cuadrático medio 22% más chico que el que obtuvimos del segundo mejor modelo que es Regresión Lineal. Además, es donde las variaciones de las variables dependientes esta explicada de la mejor manera, en un 84%, por la variación de las independientes, frente a 73% de 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 metros cuadrados de las propiedades de Capital Federal.

Adicionalmente, se puede plantear la posibilidad de buscar nuevas variables para el problema como puede ser M2*Antigüedad o transformar alguna ya existente como puede ser elevar alguna de ellas al cuadrado.

7 REFERENCIAS

- Libro Python Machine Learning (Autor: Sebastian Raschka Vahid Mirjalili)
- PYTHON APLICACIONES PRÁCTICAS (Autor: NOLASCO VALENZUELA, JORGE SANTIAGO)
- Python fácil (Autor: PÉREZ CASTAÑO, Arnaldo)